Универзитет Црне Горе
Електротехнички факултет

Славко Ковачевић

УМЕТАЊЕ ВОДЕНОГ ЖИГА УПОТРЕБОМ ДУБОКОГ УЧЕЊА

– МАГИСТАРСКИ РАД –

Подгорица, 2021. године

ПОДАЦИ И ИНФОРМАЦИЈЕ О МАГИСТРАНДУ

Име и презиме: **Славко Ковачевић**
Датум и мјесто рођења: **25. април 1996. године, Подгорица**
Назив завршеног основног студијског програма и година завршетка студија: **Електроника, телекомуникације и рачунари, 2018. године**
Назив завршеног специјалистичког студијског програма и година дипломирања: **Електроника, телекомуникације и рачунари, смјер Рачунари, 2019. године**

ИНФОРМАЦИЈЕ О МАГИСТАРСКОМ РАДУ

Назив магистарских студија: **Постдипломске магистарске академске студије, одсјек Електроника, телекомуникације и рачунари, смјер Рачунари**
Наслов рада: **Уметање воденог жига употребом дубоког учења**
Факултет/Академија на којем је рад одбрањен: **Електротехнички факултет Подгорица**

ОЦЈЕНА И ОДБРАНА МАГИСТАРСКОГ РАДА

Датум пријаве магистарског рада: **22. октобар 2020. године**
Датум сједнице Вијећа на којој је прихваћена тема: **23. фебруар 2021. године**
Ментор: **Проф. др Игор Ђуровић**
Комисија за оцјену теме и подобности магистранда:

1. **Проф. др Игор Ђуровић**
2. **Проф. др Слободан Ђукановић**
3. **Проф. др Весна Поповић-Бугарин**

Комисија за оцјену рада:

1. **Проф. др Слободан Ђукановић** (предсједник)
2. **Проф. др Игор Ђуровић** (ментор)
3. **Проф. др Весна Поповић-Бугарин** (члан)

Комисија за одбрану рада:

1. **Проф. др Слободан Ђукановић** (предсједник)
2. **Проф. др Игор Ђуровић** (ментор)
3. **Проф. др Весна Поповић-Бугарин** (члан)

Датум одбране: **27. октобар 2021. године**

Име и презиме аутора: Славко Ковачевић

Етичка изјава

У складу са чланом 22 Закона о академском интегритету и чланом 24 Правила студирања на постдипломским студијама, под кривичном и материјалном одговорношћу, изјављујем да је магистарски рад под насловом:

„Уметање воденог жига употребом дубоког учења"

моје оригинално дјело.

Подносилац изјаве:

_____

Славко Ковачевић

Подгорица, 27. октобар 2021. године

Предговор

Уметање воденог жига је поступак у којем се дигитални подаци обиљежавају воденим жигом ради очувања њихових ауторских права и аутентичности. Технике уметања воденог жига у дигиталне сигнале појавиле су се деведесетих година прошлог вијека. Сада, као и тада, највише пажње посвећује се дигиталној слици. Првобитно, методе за уметање воденог жига у аудио сигнале биле су мотивисане техникама за уметање воденог жига у дигиталну слику. Како ове технике нису могле да се у потпуности одазову проблему, предложене су бројне друге које се ослањају на специфичности аудио сигнала. Након двије деценије активног истраживања, традиционални приступи уметању воденог жига у аудио сигнале готово да су исцрпљени.

Појавом софистицираних метода за генерисање обмањујућих и лажних информација, ауторска права и интегритет информација су под озбиљном пријетњом. Испред свих типова аудио сигнала изабран је говор јер представља основни облик изражавања. Раст популарности дубоког учења довео је до његове злоупотребе, а такозвани „дипфејкови" постају све аутентичнији.

Ово истраживање које користи дубоко учење представља освјежење у области уметања воденог жига и настоји да одговори савременим проблемима. Дубока мрежа има задатак да уметне водени жиг на непримјетан начин, али и да обезбиједи робустност на нападе. Из остварених резултата може се закључити да је истраживање успјешно и да је оправдало постављена очекивања.

Систем за уметање воденог жига у говорне сигнале који ће бити презентован у овом раду представља дио већег истраживања на којем сам радио као дио тима који пред мене чине професор Игор Ђуровић и мр Коста Павловић, којима се срдачно захваљујем на сарадњи, савјетима и пруженој помоћи у рјешавању проблема.

Сажетак

Овај рад презентује систем за уметање воденог жига употребом дубоког учења. Систем се садржи од двије дубоке мреже, уметача и детектора, које имају опречне задатке и понашају се као супарници. Уметач се понаша као аутоенкодер са задатком уметања воденог жига тако да се оствари његова потпуна непримјетност. Детектор настоји да екстрахује водени жиг из сигнала носиоца. Систем као цјелина настоји да превазиђе све нападе који се могу извршити над сигналом који садржи водени жиг. Како би се остварио оптималан резултат, уметач и детектор се тренирају у пару чиме се осигурава да обје мреже остваре задовољавајуће резултате и постигну компромис. Како би се обезбиједила отпорност на нападе, одређен број улазних података се за вријеме тренирања напада. За валидацију резултата кориштене су двије метрике, непримјетност и робустност. Предложени систем остварује SNR изнад 25 dB и PESQ од 4.13 што представља добре резултате у погледу непримјетности јер су разлике између оригиналног и реконструисаног сигнала занемарљиве. У случају робустности систем остварује BER вриједности знатно испод 1% што је упоредиво са најсавременијим приступима у области уметања воденог жига.

Кључне ријечи: водени жиг, говорни сигнали, напади на водени жиг, дубоке неуралне мреже, аутоенкодери

Abstract

This paper presents a watermark embedding system that utilizes deep learning. The system consists of two deep networks, an embedder and a detector, which have opposing tasks and act as adversaries. The embedder behaves as an autoencoder and has the task of embedding the watermark so that it is completely imperceptible. The detector attempts to extract the watermark from the carrier signal. The system as a whole seeks to overcome all attacks that can be carried out on a signal containing a watermark. To achieve the optimal result, the embedder and the detector are trained jointly, which ensures that both networks achieve satisfactory results and reach a compromise. To achieve robustness to attacks, a certain amount of input signals is attacked during training. Two metrics, imperceptibility, and robustness were used to validate the results. The proposed system achieves an SNR above 26 dB and a PESQ of 4.13 which are good results in terms of imperceptibility as the differences between the original and reconstructed signal are negligible. In the case of robustness, the system achieves BER values well below 1%, which is comparable to the state-of-the-art approaches in the field of watermark embedding.

Keywords: watermark, speech signals, watermark attacks, deep neural networks, autoencoders

Садржај

1 Увод и досадашња истраживања у области

Дигитализација свих начина комуникације донијела је значајна побољшања у поновној употреби дигиталних података, њиховом складиштењу, преписивању, али и значајно олакшала њихово прикупљање и обраду. Развојем од аналогних видео-трака и компакт дискова па све до тврдих дискова и Интернет облака, технике складиштења података значајно су узнапредовале и постале приступачне крајњем потрошачу. Како се приступачност подацима у дигиталном облику повећавала, упоредо се повећавала доступност и софтверима за обраду дигиталних података који су временом постали мање сложени, лакши за употребу и све више способнији за обраду оригиналних снимака. Ови алати не представљају опасност по дигиталне податке и не би се ни на који начин смјели тако посматрати, јер су окосница дигиталне револуције и њихов развој убрзава дигитализацију друштва, али њихова злоупотреба може проузроковати несагледиве последице не само за интелектуалну својину већ и за цијело друштво. Огроман раст Интернета у последње двије деценије је од мрежне пиратерије података направио готово незаустављив тренд, а по свему судећи не постоји довољна међународна политичка воља да се овој појави стане на пут. Са правом се може рећи да се ради о проблему првог свијета. Друштва у развоју немају развијену свијест о пиратерији, али ни луксуз да о њој воде бригу. Једна од најмногољуднијих земаља свијета, Индија, недавно је легализовала фотокопирање литературе у образовне сврхе. Развијене земље су у доброј мјери успјеле да се изборе са овим проблемом, али њихове софтверске компаније које покривају глобално тржиште неријетко поспјешују пиратерију како би се изборили за што већу присутност. Аутори и власници права нису у могућности да спријече злонамјерне појединце и организације у експлоатацији њихове ауторске својине, што доноси значајне губитке у већем броју индустрија које свој профит остварују дистрибуцијом и препродајом дигиталних мултимедијалних производа. У овом раду покушано је сачувати интелектуалну својину и интегритет снимака људског говора. Говор је одабран од свих других мултимедијалних података јер се ради о основном облику изражавања, а слобода изражавања није само камен темељац демократије већ основно људско право уписано у члану 19 Универзалне декларације о људским правима[1].

Уметање воденог жига је поступак у којем се дигитални подаци обиљежавају воденим жигом ради очувања ауторских права и аутентичности тих података. Водени жиг који се умеће у сигнал носилац може бити било каква

[1]Генерална скупштина уједињених нација, резолуција Генералне скупштине A/RES/217

дигитална порука. Значење ове поруке не мора нужно бити смислено већ она може бити било која секвенца информација позната њеном аутору односно лицима која врше уметање. Технике уметања воденог жига никако не смију угрозити квалитет информације у коју се умећу јер би се тиме проблем заштите ауторских права замијенио проблемом квалитета. Изузеци су само они случајеви у којима се присуство воденог жига жели јасно нагласити, а то обично бива у домену дигиталних фотографија. Водени жиг се тада позиционира на фотографији тако да га је тешко уклонити, али се он сам обично чини транспарентним чиме се достиже циљ да је његово присуство очигледно, а оштећење информације минимално. Без обзира на то да ли ће водени жиг бити видљив или не, технике за уметање воденог жига морају постићи компромис између квалитета података и робустности воденог жига. У идеалном случају, водени жиг би требао бити непримјетна и неизбрисива порука уметнута у сигнал без губитка квалитета.

Технике уметања воденог жига у дигиталне сигнале појавиле су се деведесетих година прошлог вијека, а побројане су и детаљно анализиране у једном од најзначајнијих прегледних радова [1] те деценије у којем су у фокусу технике које се ослањају на широки спектар. У међувремену, технике уметања воденог жига значајно се унапређују и проширују на све типове дигиталних медија [2, 3]. Највише пажње посвећено је дигиталној слици [4–11], а технике које су развијене за ту употребу користиле су се за друге медијуме као што је видео [12].

Убрзо затим су се појавиле методе уметања воденог жига у аудио сигнале, а први помаци били су инспирисани већ постојећим техникама. Како ове технике нису могле да се у потпуности одазову проблему, предложене су бројне друге које су се ослањале на специфичности аудио сигнала. У овом раду усвојена је терминологија и начин класификације техника уметања воденог жига у аудио сигнале из прегледног рада [13] јер на свеобухватан и модеран начин врши испитивања постојећих техника. Постоји неколико начина за класификацију техника уметања воденог жига. Критеријуми су некада јасни и технике се могу успјешно раздвојити, али у неким случајевима долази до преплитања. Међутим, основни критеријум јесте домен у којем се врши уметање воденог жига. Очигледно је да је временски домен основни кандидат у избору домена за уметање воденог жига, а у том контексту већина техника се класификује као методе засноване на одјеку [14–16] или на временски усклађене [17–20] што је класификација из [13]. Временски домен омогућава једноставну обраду сигнала, али има своје мањкавости прије свега у погледу робустности воденог жига. Робустна техника уметања воденог жига у временском домену [17] ко-

ристи биполарну секвенцу као водени жиг која се генерише помоћу Рењијеве (енгл. *Renyi*) мапе [21]. Уметање воденог жига врши се у сегментима оригиналног сигнала који се не преклапају. Снага воденог жига се фино подешава како би се осигурала нечујност.

У циљу унапређења робустности, научна истраживања су пребјегла употреби трансформационих домена. Техника широког спектра предложена је у [1] и односи се на све мултимедије те се може примијенити и у аудио сигналима. Ова техника је накнадно развијана и уподобљена специфично за дигиталне аудио сигнале [22–26]. Квантизациона модулација индекса (енгл. *Quantization Index Modulation - QIM*) је још један припадник класе метода које се ослањају на трансформационе домене. Ова техника се састоји од два кључна корака, индекс модулације и квантизације, а представљена је у [27]. Оригинално, употријебљена је за уметање воденог жига у домену дигиталне слике. Идеја из овог рада преточена је у домен аудио сигнала и искориштена у бројним другим радовима [28–33]. Печворк алгоритам (енгл. *Patchwork algorithm*) предложен у [34] нашао је примјену у уметању воденог жига у аудио сигнале [35]. Модификација овог алгоритма [36] побољшава робустност и непримјетност воденог жига. Као подлога за поменуте приступе обично се користе коефицијенти добро познатих ортогоналних трансформација као што је дискретна Фуријеова трансформација (енгл. *discrete Fourier transform - DFT*) [37], дискретна вејвлет трансформација (енгл. *discrete wavelet transform - DWT*) [28, 33] [26, 31, 38] или дискретна косинусна трансформација (енгл. *discrete cosine transform - DCT*). Техника која као домен користи DCT предложена је у [31] и врши пажљиво распршивање енергије воденог жига како би се постигао компромис између снаге и нечујности воденог жига. Више метода ослоњених на DWT су предложене у [33]. Биполарни кодови за синхронизацију и битови воденог жига умећу се у различите подопсеге DWT у оквирима које одређује праг интезитета. DFT, фракциона Фуријеова трансформација (енгл. *fractional Fourier transform - FrFT*) и кватернион DFT су имплементиране и упоређене у [37] у односу на домен уметања воденог жига. Значајна разлика у перформансама и робустности није уочена. Међутим, трансформациони домени се у доброј мјери комбинују [29, 30, 39] или се могу подједнако користити у примјени [36]. Из тог разлога, трансформациони домени нису увијек доминантан дио технике те се самим тим у таквим случајевима ни не могу користити за подјелу приступа. Уметање воденог жига гдје се DWT и DCT комбинују је предложено у [39]. DWT се користи како би се избациле нископропусне компоненте сигнала у који је водени жиг уметнут помоћу DCT.

Машинско учење се такође користи у области водених жигова и то прије

свега за његово откривање унутар сигнала носиоца [40]. Раст популарности дубоких неуралних мрежа (енгл. *deep neural network - DNN*) није заобишао ни област уметања воденог жига. Значајан број новијих радова користи DNN не само за откривање тј. детекцију воденог жига већ и за његово уметање [41–45]. Међутим, технике базиране на DNN и даље касне за традиционалним приступима. За разлику од других области дигиталне обраде сигнала гдје је вјештачка интелигенција направила огромну разлику и значајно убрзала напредак, област воденог жига и даље не профитира од неуралних мрежа. Иронично, DNN се већ користе у овој области, али на супротној страни гдје остварују завидне разултате. Наиме, генерисање обмањујућих и лажних информација никада није било лакше, укључујући појаву такозваних дипфејкова (енгл. *deepfake*) који представљају лажи генерисане помоћу дубоких мрежа. Ови напади се не користе само за ширење лажних вијести већ и за повреду части и угледа појединаца или институција, а могу се чак и искористити за крађу идентитета. Последице таквог дјеловања су огромне, а у случајевима људског говора могу чак довести и до губитка живота чему су подложне оружане снаге и све друге организације у којима се наређења и информације преносе говором. За очекивати је да ће се овој појави бити могуће супроставити користећи исте технике, али на потпуно супротан начин. Како би се у томе успјело, неопходно је пронаћи нове технике уметања воденог жига које користе иста средства, али су у стању да се успјешно одбране од дипфејкова.

Приступ предложен у овом раду користи DNN за уметање и детекцију воденог жига у аудио сигнале. Ова два задатка су у супротности те их систем мора довести у равнотежу. Водени жиг би требао бити робустан и отпоран на нападе, али квалитет сигнала носиоца не смије се нарушити. Из тог разлога, овај рад предлаже систем који се састоји од двије цјелине. Први дио система, уметач, за задатак има да уметне водени жиг у оригинални сигнал. Како се за израду овог система користе DNN, традиционални домени се неће користити за репрезентацију сигнала, мрежа уметача генерише трансформациони домен који представља димензионо компримовану верзију оригиналног сигнала. Овај латентни простор зависи од улазних података који се користе у процесу тренирања мреже. Као улаз у мрежу користе се сирови подаци, временски одбирци сигнала говора. Ову одлуку оправдава чињеница да се DNN користе за екстракцију карактеристика. Други дио система, детектор, за задатак има да детектује односно екстрахује водени жиг у сигналу који га садржи. Понаша се као класификатор јер број порука које представљају водени жиг није бесконачан већ ограничен. Ове двије неуралне мреже су противници јер су им и циљеви у супротности. Како би читав систем могао да

оствари успјех, потребно је да обје мреже задовоље своје циљеве. Оптимално рјешење је оно које представља компромис. Заједнички и пажљиво спроведен поступак тренирања омогућава конвергенцију обје мреже. То подразумијева да је порука довољно добро сакривена тако да је детектор може пронаћи, али да њена постојаност унутар носиоца не нарушава квалитет информације. Ово представља идеалан случај. Систем се мора дизајнирати тако да се у разматрање узме и могућност напада на сигнал који могу утицати на присутност воденог жига или на његов квалитет. Техника предложена у овом раду тестирана је на скупу говорних сигнала у односу на остале приступе у погледу квалитета сигнала и робустности система.

Способност предложене технике да се успјешно одбрани од реверзног инжењеринга је веома важна. Уколико систем не би био у могућности да се заштити од ове врсте напада, малициозни појединци би били у могућности да у потпуности компромитују читав процес. Сама комплексност система и употреба DNN чини обрнути инжењеринг готово немогућим. Повећавањем комплексности смањују се и перформансе система. Перформансе уметача нису од значаја јер његови резултати нису потребни у реалном времену. Ово не важи за детектор који мора бити што бржи, а примјену потенцијално може наћи и на хардверски скромним рачунарима, мобилним уређајима и сл. Због тога је детектор сведен на једноставан класификатор који не захтијева интензивне прорачуне.

Рад је организован на следећи начин. Поглавље 2 дефинише начин уметања воденог жига помоћу трансформационих домена, али и неке од познатих трансформација чије особине су утицале да се од употребе трансформационог домена одустане. Поглавље 3 прелази неколико напада који су у сврху доказивања робустности система имплементирани и кориштени како би се симулирао сценарио у којем сигнал може бити нападнут или оштећен. Неуралне мреже односно њихове основе које су неопходне да би се разумио рад система описане су у поглављу 4. Архитектура система, мреже уметач и детектор до детаља су објашњени у поглављу 5. Процес тренирања демонстриран је у поглављу 6. У поглављу 7 представљен је корпус података који ће се користити за тренирање неуралне мреже и валидацију њених резултата. Како би се успјех овог истраживања истакао на прави начин, потребно је јасно и недвосмислено дефинисати методе за мјерење перформанси уз помоћ којих ће се остварени резултати упоредити са резултатима других радова из ове области. Методе за мјерење перформанси као и резултати алгоритма презентовани су у поглављу 8. Закључак и потенцијална будућа истраживања дати су у поглављу 9.

2 Домени за уметање воденог жига

Уметање воденог жига у временском домену се може представити као:

$$y(n) = x(n) + \alpha w(n) \tag{1}$$

при чему y представља сигнал у који је уметнут водени жиг, док су x и w оригинални сигнал, односно водени жиг. Параметар α одређује снагу воденог жига чиме утиче на његову чујност, али и на могућност детекције. Овдје је дат као константан, али уопштено се може рећи да он варира и да се прилагођава садржају сигнала све док се не успостави стање у којем је водени жиг мање чујан, али лак за откривање. Имплементације уметања воденог жига у временском домену су ефикасне за реализацију, али је робустност на нападе неадекватна због чега се прибјегава трансформационим доменима. Уметање воденог жига у неком трансформационом домену се може представити као:

$$Y(k) = X(k) + \alpha W(k) \tag{2}$$

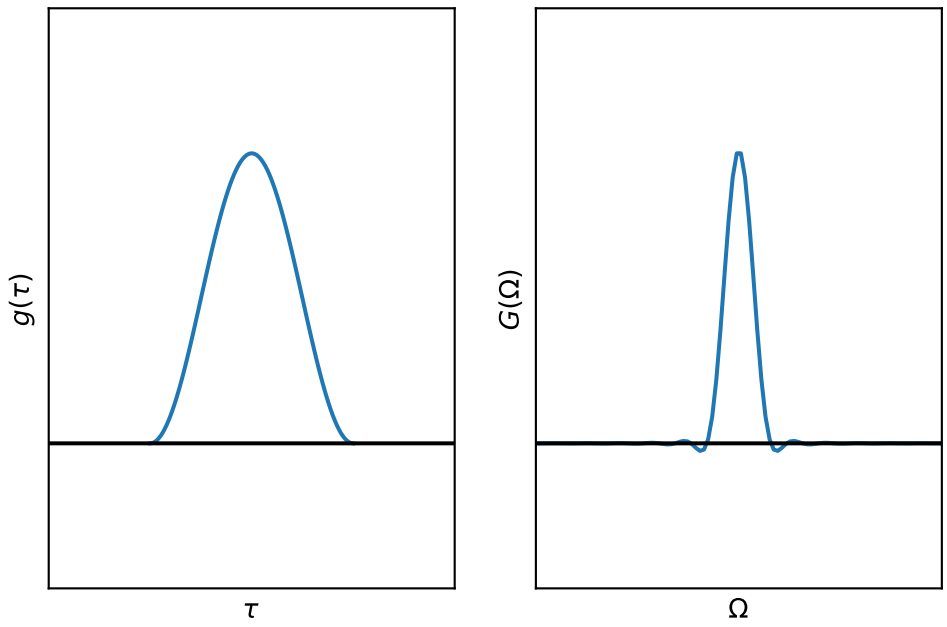
гдје Y представља репрезентацију сигнала у који је уметнут водени жиг у трансформационом домену, док су X и W оригинални сигнал и водени жиг у том истом домену. Постоји неколико трансформационих домена који се користе за уметање воденог жига, а неки од њих ће бити представљени у наредним поглављима.

2.1 Краткотрајна Фуријеова трансформација

Краткотрајна Фуријеова трансформација (енгл. *Short-time Fourier transform - STFT*) представља временско-фреквенцијску репрезентацију сигнала. Употребом клизајућих прозор функција $g(t)$ локализује се Фуријеова трансформација над сигналом $x(t)$. Пролазећи прозор функцијом врши се Фуријеова трансформација само оног дијела сигнала који се у датом тренутку налази унутар покретног прозора чиме се добијају спектралне компоненте тог дијела сигнала, а што је условљено ширином прозора и кораком помјерања. Дискретни облик STFT је [46]:

$$STFT(n,k) = \sum_{m=-N/2}^{N/2-1} g(m)\, x(n+m)\, e^{-j2\pi mk/N} \tag{3}$$

гдје је g прозор којим се локализује сигнал у временско-фреквенцијској равни. N је дужина прозора. Квалитет STFT умногоме зависи од избора функције



Слика 1: Ханов прозор у временском (лијево) і фреквенцијском (десно) домену

прозора, њене дужине, али и корака помјерања. Постоји више типова функције прозора, а Ханов прозор је једна од најчешћих у употреби. Она има облик:

$$g(t) = \begin{cases} 0.5\left(1 + \cos\left(\pi\tau/T\right)\right) & \text{за } |t| < T \\ 0 & \text{остало} \end{cases} \tag{4}$$

И може се видјети на слици 1. Правоугаони прозори нуде једнаку репрезентативност свим одбирцима зато што је вриједност функције константна, али се у фреквентном домену јављају осцилаторни ефекти који су нежељени. Прозори као Ханов рјешавају проблем у фреквентном домену, али уносе други проблем у временском домену. Због тога што њихова вриједност није константна, долази до скалирања одбирака приликом рачунања STFT, а најбоље пролазе они одбирци који су ближи тренутку посматрања. Због тога се уводе преклапајући прозори. Ширина ових прозора не може се лако одредити. Временски широк прозор нема могућност детекције брзих промјена у времену, али има високу фреквенцијску резолуцију. Узан прозор детектује брзе промјене у времену, али има малу фреквенцијску резолуцију. Самим тим, погрешан избор ових параметара може да унесе грешку у читав систем од старта и онемогући неуралној мрежи да на ваљан начин изврши обраду података.

2.2 Спектрограм

Спектрограм представља квадрирану апсолутну вриједност STFT и дефинише се као:

$$SPEC\,(n,k) = |STFT\,(n,k)|^2 \tag{5}$$

и може се рећи да је то енергија STFT. Очигледан недостатак спектрограма је то што се инверзна трансформација не може једноставно извршити, а губици у квалитету су неизбјежни. Једини начин да се до почетне информације дође јесте помоћу естимације [47, 48].

2.3 Мел спектрограм

Мел спектрограми покушавају моделовати људски слух. То раде тако што на периодограму сигнала употребљују банку филтара на спектру снаге и сумирају енергије по филтру. Ове вриједности се касније логаритмују, а над њима се рачуна DCT. Филтри из банке филтара нису линеарни, њихова ширина расте како расте и фреквенција, а међусобно се преплићу. Тиме се покушава имитирати људски слух који порастом фреквенције постаје мање толерантан на промјене. Овако добијена репрезентација сигнала корисна је само у случајевима када инверзна трансформација није потребна систему за даљи рад јер је у овом случају у потпуности немогућа.

2.4 Алтернатива трансформационим доменима

Из досадашњих поглавља јасно је да трансформациони домени могу у доброј мјери да помогну систему у остваривању циља, али захтијевају додатну пажњу и доменско знање. У случају рачунања STFT, што условљава и рачунање спектрограма, али и у рачунању Мел спектрограма, одређени параметри као што је избор прозор функције, њена ширина и корак помјерања играју значајну улогу. Због тога се са правом може сумњати да ће DNN потенцијално прескочити неку вриједну информацију о сигналу која лошим избором трансформационог домена и његових параметара може бити бесповратно изгубљена. Такође, поставља се и питање реконструкције оригиналног сигнала што није увијек могуће. Из угла DNN, оне су замишљене тако да раде са сировим подацима. Њихов основни задатак јесте екстракција корисних информација из улазних података на основу којих се стиче знање о подацима и рјешава проблем. Овакав приступ може да донесе значајне предности, али

и да унесе нестабилност. Наиме, дозвољавајући мрежи да се стара о проналажењу подобне репрезентације улазних сигнала успорава се учење и отвара могућност неуспјеха односно дивергирања у процесу учења.

3 Напади на водени жиг

Идеални услови у којима систем за уметање и детекцију воденог жига постиже добре резултате нису довољни да би се систем као такав сматрао успјешним. У реалним условима, водени жиг ће бити оштећен у одређеној мјери. До тога може доћи на неколико начина. Злонамјерно руковање сигналом подразумијева нападе на сигнал са циљем да се он у потпуности обрише чиме би се угрозила нечија интелектуална својина. У случају говора ово само по себи нужно није проблем јер је заштита тачности информације важнија од заштите интелектуалне својине која је недвосмислена, а у многим случајевима ради се о јавно доступним говорима који сами по себи не подлијежу ауторским правима. Потпуни нестанак воденог жига може се сматрати доказом малверзације само уколико систем за детекцију воденог жига у свим осталим случајевима оштећења који су производ валидних обрада сигнала успјешно детектује водени жиг. Промјена фреквенције одабирања, скраћивање снимка, уклањање тишина, компресија и сл. су примјери добронамјерне обраде говорног сигнала и не морају нужно представљати малициозну радњу. Међутим, са аспекта система за уметање и детекцију воденог жига, све ове операције било да су добронамјерне или не, сматрају се нападима. Уколико систем и након ових операција успијева да детектује водени жиг сматра се робустним. Робустност представља највећи изазов за систем због обиља напада и њихове софистицираности.

До сада је предложен значајан број напада на системе за уметање и детекцију воденог жига. Неки од прегледних радова о нападима су [49–51], али ниједан од њих не покрива све нападе. Ово је оправдано јер број категоризованих и забиљежених напада не представља нужно све могуће нападе. Аутори [49] предлажу референтни скуп напада (енгл. *benchmark*) који ће се користити за тестирање система за уметање и детекцију водених жигова односно као мјера њиховог квалитета. Појавом нових технологија на пољу дигиталне обраде сигнала за очекивати је да ће се појавити и нови напади. Као што је већ речено, DNN односно машинско учење као шири појам, остварује значајне резултате у развоју напада. Осмишљање нових напада је активно подруче истраживања.

Аутори радова који покушавају остварити добре резултате у робустности система морају се одлучити за референтни скуп података или пак изабрати одређен број напада против којег ће се борити њихов систем. Њихов задатак је да на тако ограниченом скупу успију да учине да је њихов систем у потпуности отпоран на дату врсту напада. Аутори [33] су тестирали предложени

систем на великом броју напада. Нажалост, ниједан систем није отпоран на све нападе јер се за вријеме његовог дизајнирања не могу узети у обзир сви напади зато што је њихов број потенцијално бесконачан. Класификацијом напада у неколико група које сачињавају напади сличне природе те одабиром одређеног броја напада из сваке од група повећале би се шансе система у одбрани од напада над којим није до сада трениран.

У овом истраживању гдје уметач и детектор чине један систем, одбрана од напада не може бити задатак искључиво детектора без обзира што се напад може догодити тек онда када сигнал напусти мрежу уметача. Наиме, уколико уметач не изврши уметање воденог жига на начин који је отпоран на нападе, неки од разматраних напада може уништити водени жиг прије него ли он дође до детектора те самим тим онемогућити детектору да изврши детекцију. Примјера ради, уколико уметач научи да водени жиг смјешта само у одређеном опсегу фреквенција, филтрирањем сигнала извршило би се брисање воденог жига. Уметач мора да научи да водени жиг развуче дуж читавог спектра фреквенција. Ова два дијела система се дакле морају заједно тренирати и то за вријеме напада како би заједничким снагама научили како да одбране систем. Све грешке које детектор направи морају се пропагирати назад ка уметачу. Како би се олакшала практична реализација оваквог система, напади се имплементирају као слојеви неуралних мрежа.

Три главне групе напада биће разматране у овом раду. Оне су филтрирање, десинхронизација и уметање шума које може покрити све врсте напада изазваних губицима од компресије. За представнике ових група изабрани су Батервортов филтар, пригушивање узорака и адитивни Гаусов шум. Сви ови напади имају одређене параметре, који су бирани пажљиво. Важно је да њихов утицај не поквари перцептивни квалитет сигнала. Уколико би се десило да су напади толико деструктивни да униште саму информацију, они губе смисао.

3.1 Батервортов филтар

Нископропусни филтри се често користе у нападима на водени жиг. У случају људског говора, већина информација налази се на ниским фреквенцијама чиме се оставља простор да се водени жиг упише на високим. Овај наиван приступ тешко подноси нископропусно филтрирање при којем се водени жиг може поремити или у потпуности обрисати. Батервортов филтар је врста дигиталног филтра дизајнираног да умањи таласање фреквенцијског одзива филтра. Може се назвати и максимално равним филтром јер има најстрмију

преносну функцију за одређени степен без изазивања таласања у осталим дијеловима фреквенцијског одзива. Сматра се компромисом између Чебишевог и Беселовог филтра. Амплитуда одзива Батервортовог филтра дата је као:

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2N}}} \tag{6}$$

гдје ω представља угаону фреквенцију, ω_c и N су два параметра помоћу којих се дефинише филтар. ω_c је гранична фреквенција, а N је ред филтра. Амплитуда одзива Батервортовог филтра шеснаестог реда са граничном фреквенцијом 4 kHz је дата на слици 2а.

Полови функције преноса Батервортовог филтра $H(s)$ у s домену налазе се у кругу полупречника ω_c и дати су са:

$$s_k = \omega_c \cdot e^{\frac{j\pi(2k-1+N)}{2N}} = \omega_c \cdot \left[\cos\left(\frac{\pi(2k-1+N)}{2N}\right) + j\sin\left(\frac{\pi(2k-1+N)}{2N}\right)\right] \tag{7}$$

за $k \in \{1, 2, ..., N\}$. Након одређивања полова, преносна функција Батервортовог филтра се дефинише на следећи начин:

$$H(s) = \frac{1}{(s - s_1) \cdot (s - s_2) \cdots (s - s_N)} = \sum_{k=1}^{N} \frac{r_k}{s - s_k} \tag{8}$$

гдје r_k представљају коефицијенте добијене парцијалном декомпозицијом преносне функције $H(s)$. Како би се овај филтар примјенио у временском домену, неопходно је израчунати његов импулсни одзив. Слика 2б приказује импулсне одзиве Батервортовог филтра различитог реда. Импулсни одзив Батервортовог филтра се дефинише као:

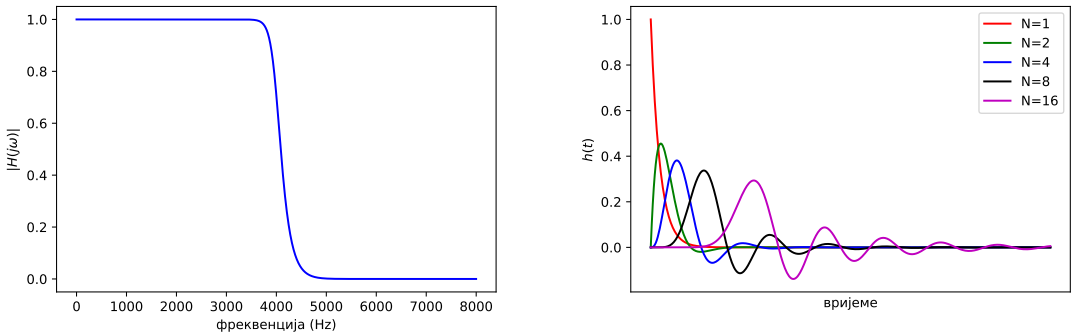
$$h(t) = \sum_{k=1}^{N} r_k \cdot e^{s_k t} \cdot u(t) \tag{9}$$

гдје s_k представљају полове израчунате у (7), док је $u(t)$ Хевисајдова функција, а r_k су коефицијенти израчунати у (8). Имплементација импулсног одзива извршена је у Пајтон програмском језику[2].

Након рачунања импулсног одзива, филтрирање се врши конволуцијом између сигнала и импулсног одзива:

$$y(t) = (x * h)(t) \tag{10}$$

[2]www.github.com/skovacevic96/deepsignal/blob/main/utils/butterworth.py



(а) **(б)**

Слика 2: (а) Амплитуда одзива Батервортовог филтра шеснаестог реда са граничном фреквенцијом 4 kHz ($\omega_c = 2 \cdot \pi \cdot f_c$), (б) Импулсни одзиви Батервортових филтара различитог реда

што је од великог значаја јер се једноставно може имплементирати као слој неуралне мреже. Тежине овог слоја морају бити фиксне, а сам слој не може се додатно обучавати. Његове тежине су дате једначином (9).

Дискретизована временска реализација конволуције из једначине (10) има облик:

$$y(n) = \sum_{m=-\infty}^{+\infty} x(m) \cdot h(n-m). \tag{11}$$

Како се ради о слоју неуралне мреже, неопходно је утврдити на који начин ће се кроз њега пропагирати градијенти. О томе ће више ријечи бити у поглављу 4.4. Парцијални изводи у односу на улазни сигнал x се рачунају као:

$$\frac{\partial y(n)}{\partial x(m)} = h(n-m) \tag{12}$$

и неопходни су како би се грешка пропагирала од декодера ка уметачу. Извод функције трошка E по x је:

$$\frac{\partial E}{\partial x(m)} = \sum_{n=0}^{L} \frac{\partial E}{\partial y(n)} \cdot \frac{\partial y(n)}{\partial x(m)} = \sum_{n=0}^{L} \frac{\partial E}{\partial y(n)} \cdot h(n-m). \tag{13}$$

При одабиру граничне фреквенције филтра узет је у обзир оспег фреквенција људског говора и одлучено је да је 4 kHz оптимална вриједност. Такође, експериментално је доказано да граничне фреквенције испод одабране уносе чујна оштећења.

3.2 Пригушивање одбирака

Пригушивање одбирака је један од најједноставнијих типова десинхронизујућих напада. Ова врста напада насумичне одбирке сигнала поставља на нулу. Може се дефинисати као:

$$y(n) = \text{mask}(n) \cdot x(n) \tag{14}$$

гдје маска представља насумично генерисан биполарни низ помоћу којег се дефинише који од одбирака ће бити пригушени. Дужина маске је фиксна. Парцијални изводи по x су:

$$\frac{\partial y(n)}{\partial x(m)} = \begin{cases} \text{mask}(n) & n = m \\ 0 & n \neq m. \end{cases} \tag{15}$$

По правилу извода сложене функције имамо да је:

$$\frac{\partial E}{\partial x(m)} = \sum_{n=0}^{L} \frac{\partial E}{\partial y(n)} \cdot \frac{\partial y(n)}{\partial x(m)} = \frac{\partial E}{\partial y(m)} \cdot \text{mask}(m). \tag{16}$$

Како је дужина маске константна она се мора бирати тако да не угрози квалитет сигнала. Експериментално је утврђено да је дужина од 1000 одбирака адекватна, а тај број представља око 3% дужине улазног сигнала.

3.3 Адитивни Гаусов шум

Додавање шума представља најчешћи вид напада на водени жиг. Као модел за шум искоришћен је StirMark тест [49]:

$$y(n) = x(n) + \alpha \cdot \epsilon(n) \tag{17}$$

гдје су x, ϵ и y оригинални сигнал, шум и излазни сигнал док је α параметар који дефинише релативну снагу шума у односу на оригинални сигнал. За вриједност α изабрана је 0.009 како се не би значајно погоршао квалитет сигнала. То поткрепљује чињеница да је сада резултујући однос сигнал-шум у просјеку 30 dB. Шум се додаје како би се покушала уништити порука која се за вријеме уметања умеће у сигнал носилац. Додавање шума представља најнаивнији приступ представљен у овом поглављу.

Као и при дефинисању осталих напада, сви параметри су фиксни и не могу се мијењати за вријеме тренирања. У овом случају ти параметри су снага шума и његова дистрибуција. Из тога се може закључити да се само

градијент по улазном сигналу x мора пропагирати слојевима неуралне мреже који се налазе иза овог слоја. Сви остали су једнаки нули. Парцијални изводи y по x се дефинишу као:

$$\frac{\partial y(n)}{\partial x(m)} = \begin{cases} 1 & n = m \\ 0 & n \neq m. \end{cases} \tag{18}$$

По правилу извода сложене функције, укупан трошак E који се пропагира је:

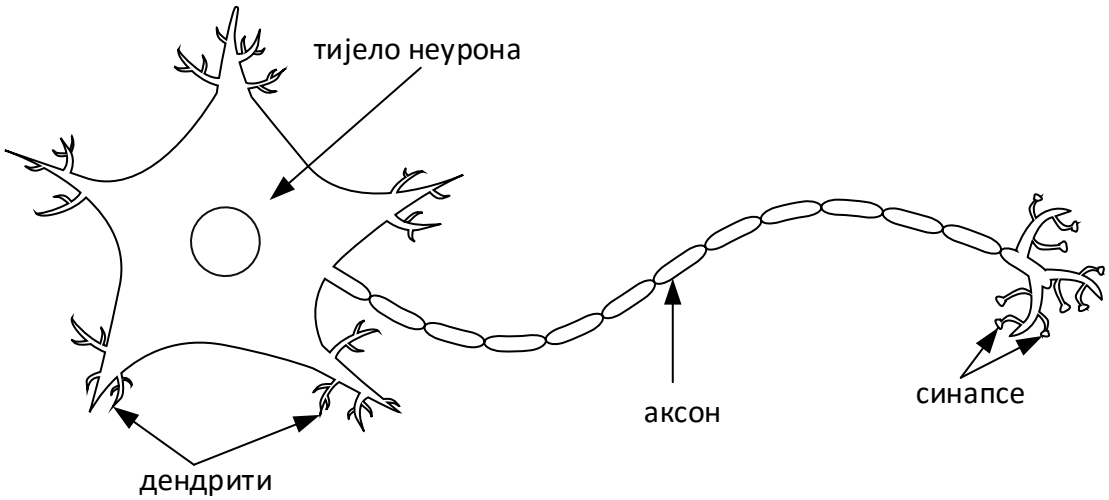
$$\frac{\partial E}{\partial x(m)} = \sum_{n=0}^{L} \frac{\partial E}{\partial y(n)} \cdot \frac{\partial y(n)}{\partial x(m)} = \frac{\partial E}{\partial y(m)} \tag{19}$$

где L представља дужину сигнала.

4 Неуралне мреже

Вјештачке неуралне (неуронске) мреже представљају подскуп машинског учења, подобласти вјештачке интелигенције, и састоје се од великог броја различитих алгоритама који се труде моделовати биолошки нервни систем. Састоје се од међусобно повезаних процесних елемената који се називају вјештачким неуронима. Показало се да је ово поређење са биолошким нервним системом у доброј мјери погрешно, али је сам назив опстао и даље се користи. Неуралне мреже се тренирају тј. обучавају. Потребно је за жељени унос задати жељени износ, а задатак мреже је да са аспекта дигиталне обраде сигнала, пронађе преносну функцију која ће бити у стању да испуни задатак. Оваква мрежа може се моделовати као скуп неурона повезаних у ацикличном графу јер излаз из једног неурона може бити улаз другог неурона. Заједничко за све неуралне мреже јесте појам архитектуре односно шеме везивања неурона. Архитектура мреже говори о томе како су неурони повезани и може се подијелити на слојеве. Изворно, неуралне мреже биле су сачињене од свега два слоја. Улазног, излазног и скривеног. Мрежа је N-слојна ако има $N-1$ скривених слојева имајући у виду да улазни слој заправо и није слој неуралне мреже јер представља податке. Развијањем хардверских могућности рачунара број скривених слојева је растао. Порастом броја скривених слојева дошло је до развијања дубоког учења, посебне подобласти неуралних мрежа. Постоји неколико типова слојева који могу чинити неуралну мрежу. Изворно, неуралне мреже имале су само потпуно повезане (енгл. *Fully connected* - FC) слојеве који су сада само један од неколико слојева у употреби. За FC слојеве важи да се излаз сваког неурона из претходног слоја повезује са улазима свих неурона наредног слоја. Не постоји веза између неурона истог слоја. Неурони обављају једну операцију, а везама између њих додјељују се тежински коефицијенти. Ови коефицијенти се за вријеме тренинга мијењају, а прије тренинга се иницијализују неком вриједношћу. Наиме, када би иницијалне вриједности биле нула, тада би било немогуће да било каква информација стигне од улаза до излаза. Како одабрати иницијалне вриједности? На ово питање не постоји једноставан и јединствен одговор већ се њиме бави посебна област неуралних мрежа која се назива иницијализација.

Вјештачки неурон није успио да оправда очекивања поређења са биолошким неуроном, али јесте моделован по њему па је за појашњење његовог рада згодно објаснити функционалност биолошког неурона. Поједностављен приказ биолошког неурона приказан је на слици 3. Неурон помоћу дендрита прима сигнале од околних неурона, обрађује их и шаље даље другим неу-

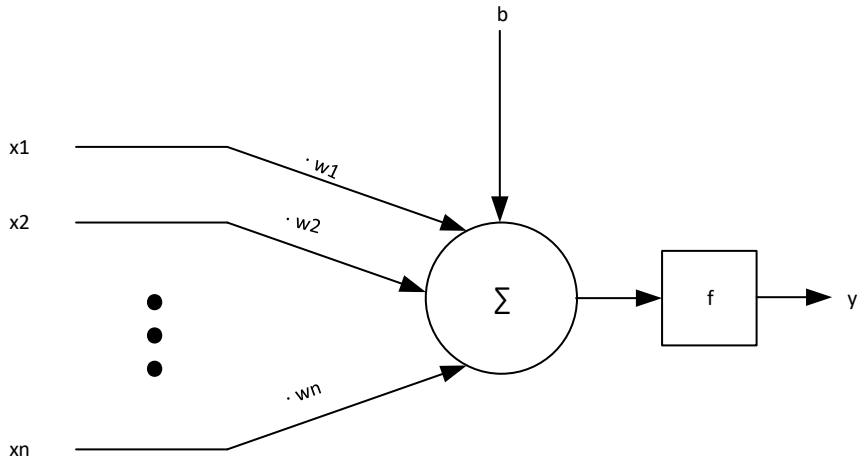


Слика 3: Поједностављен приказ биолошког неурона

ронима са којима је везан синапсама. Слање сигнала врши помоћу аксона. Вјештачки неурон може имати произвољан број улаза $x_1, x_2, ..., x_n$. Вриједности ових улаза множе се тежишним коефицијентима, а након тога сумирају како би се добила јединствена вриједност. Резултат ове суме назива се логит и записује се као $z = \sum_{i=0}^{n} w_i x_i$. Како би се пронашла функција преноса, логиту се обично додаје и одступање (*engl. bias*). Да би се овај појам боље објаснио најбоље је посматрати линеарну функцију облика:

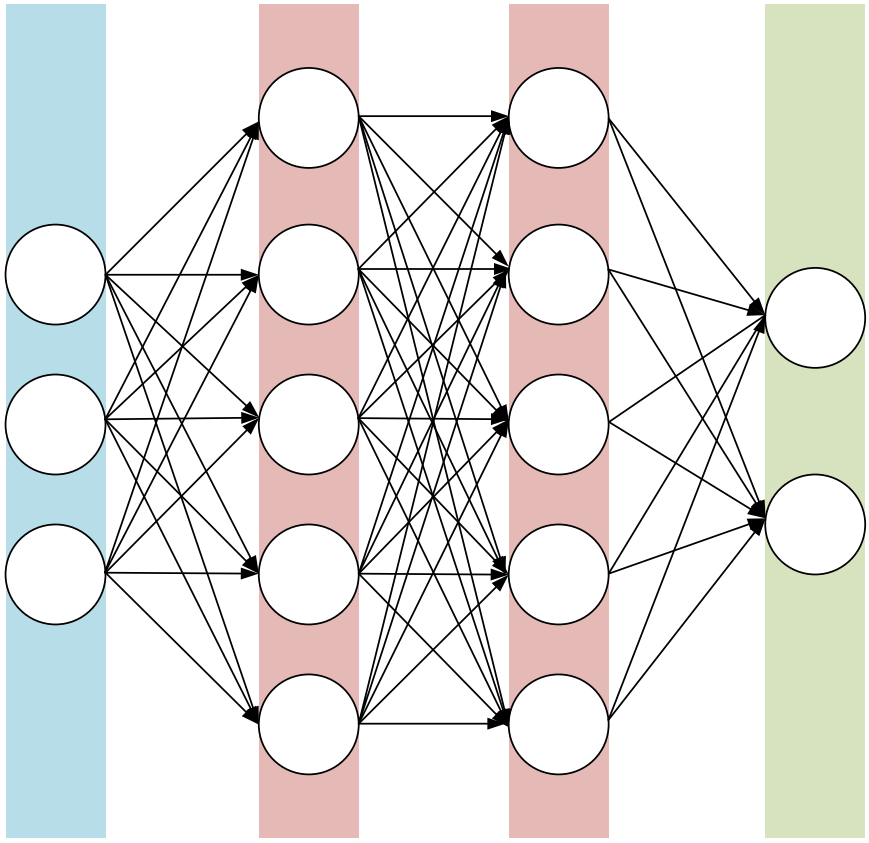
$$f(x) = y = ax + b \tag{20}$$

гдје b представља одсјечак на оси y. Јасно је да и најједноставнији задатак као



Слика 4: Блок шема неурона

што је проналазак једначине праве која пролази кроз двије тачке не би могао бити савладан без параметра b у општем случају. Одступање рјешава сличан



Слика 5: Неуронална мрежа са два скривена слоја, кружнице представљају неуроне, а линије везе између њих

проблем код неуралних мрежа односно код FC слојева који имају велики број промјењљивих, а додаје се за сваки неурон. За разлику од тежинских коефицијената, одступање се може иницијализовати нулама што се често и ради. Добијена вриједност логита пролази кроз нелинеарност коју уносе активационе функције о којима ће више ријечи бити у поглављу 4.1. Блок шема неурона приказана је на слици 4. Векторизовано имамо да су улаз слоја и тежине:

$$\boldsymbol{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \boldsymbol{W} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \tag{21}$$

па је излаз односно вриједност неурона:

$$y = f\left(\boldsymbol{W^T X} + \boldsymbol{b}\right) \tag{22}$$

гдје $f()$ представља активациону функцију.

На слици 5 се може видјети примјер неуралне мреже коју чине FC слојеви, а која има два скривена слоја. Ови слојеви имају једнак број неурона што не мора бити случај. FC слојеви се на једноставан начин могу реализовати употребом матричног записа. Тако се тежишни коефицијенти веза два слоја неуралне мреже могу представити са:

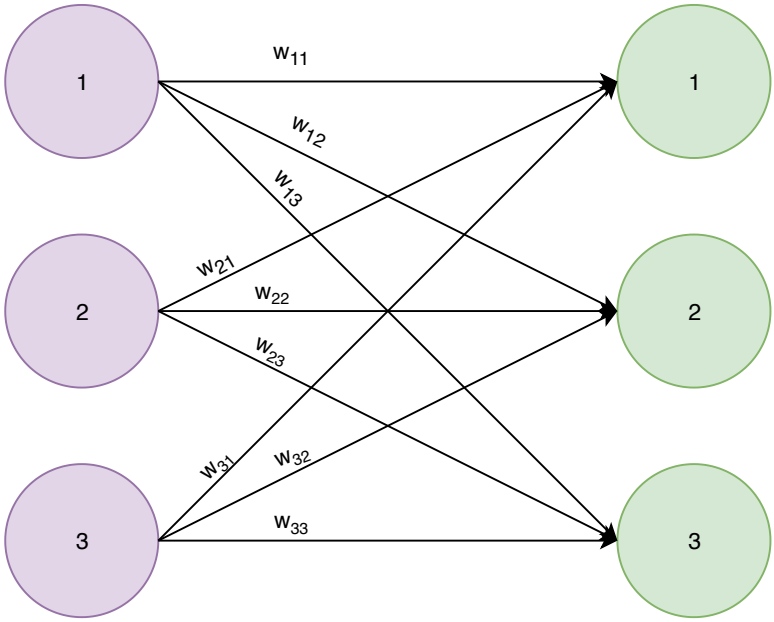
$$\boldsymbol{W} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \tag{23}$$

и приказане су на слици 6. Нека први слој има вриједности $\boldsymbol{h_1}$, а други слој $\boldsymbol{h_2}$. Вриједност свих неурона једног слоја може се такође представити једначином (22) само би овога пута y био вектор чија би дужина била једнака броју неурона које тај слој има. За други слој оне се могу израчунати на следећи начин:

$$\boldsymbol{h_2} = f\left(\boldsymbol{W} \times \boldsymbol{h_1} + \boldsymbol{b}\right) \tag{24}$$

гдје је $\boldsymbol{b}$ одступање димензија 3×1.

Помоћу неуралних мрежа у протекле двије деценије ријешено је много проблема у области дигиталне обраде сигнала. Нажалост, због њихове огромне популарности неријетко се користе за рјешавање и оних проблема који се могу ријешити традиционалним методама. Програмирање њихове архитектуре није једноставан посао. Како је развој неуралних мрежа условљен хардверским могућностима савремених рачунара, пажња се посвећује оптимизацији. Постоји



Слика 6: Тежишни коефицијенти веза два слоја неуралне мреже

неколико програмерских оквира отвореног кода који олакшавају програмирање архитектуре неуралних мрежа, а за потребе овог истраживања кориштен је Tensorflow[3]. Поред звучног имена и несумњивог успјеха, неуралне мреже су само један мали корак ка емулацији природне интелигенције и никако се не могу и не смију поредити са биолошким неуронским мрежама од којих су позајмиле назив.

Прича о неуралним мрежама биће подијељена у неколико поглавља. Поглавље 4.1 објашњава мотив за увођење активационих функција и даје преглед оних активационих функција које се користе у истраживању. Конволуциони слојеви неуралних мрежа односно конволуционе неуралне мреже користе се у предложеној архитектури и описане су у поглављу 4.2. Како би се ажурирали тежински коефицијенти, неопходно је израчунати трошак употребом функције трошка, поглавље 4.3, а о пропагацији градијената кроз слојеве биће говорено у поглављу 4.4. За стабилнији процес тренирања и вјероватнију конвергенцију задужене су оптимизација, о којој ће бити ријечи у поглављу 4.5, и нормализација, поглавље. 4.6.

4.1 Активационе функције

Активациона функција или нелинеарност вјештачког неурона за задати улаз дефинише његов излаз. Појам нелинеарност користи се оправдано јер ли-

[3]www.tensorflow.org

неарни перцептрони могу ријешити само тривијалне проблеме са малим бројем неурона. У овом раду биће разматране само оне активационе функције које мрежа користи уз пар изузетака које је неопходно увести због употпуњавања одређених поглавља. Оне су:

1. Сигмоид

Сигмоид је математичка функција која подсјећа на слово S. Сагледано математички, више функција могу бити сигмоид функције, али се у области неуралних мрежа овај израз одомаћио за логистичку функцију. Њен математички облик је:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{25}$$

и приказана је на слици 7а. Сигмоид ограничава излазне вриједности неурона и његов улаз пресликава у опсег $[0, 1]$. Јасно је да ће велике вриједности улаза бити пригушене па ће се сви негативни улази који теже ка $-\infty$ пресликати у 0 док ће позитивни улази који теже ка ∞ бити пресликани у 1. Ово својство сигмоида добро интерпретира брзину пуцања неурона што је и био мотив за њено увођење. Пуцање неурона представља се са два стања. Једно у којем се не активира и његов излаз је стално 0 и друго стање у којем је неурон у потпуности засићен и на излазу даје увијек 1. Како сигмоид ограничава излазне вриједности то га чини склоним убијању градијената чиме се тренирање зауставља.

2. ReLU

ReLU (енгл. *Rectified Linear Unit*) је активациона функција уведена са циљем да превазиђе недостатке сигмоида елиминацијом ограничавања излазних вриједности. Њен математички облик је:

$$f(x) = \max(0, x) \tag{26}$$

и представља ништа друго до линеарну функцију за вриједности улаза $x > 0$, а нулу за улазне вриједности $x \leqslant 0$. Поред предности у односу на сигмоид, ReLU активациона функција стекла је популарност и због своје једноставности. Приказана је на слици 7б. Очигледно је да ће све негативне вриједности имати на излазу 0 чиме се може успорити тренинг. Активационе функције које су училе на грешкама ReLU активационе функције покушавају ријешити овај проблем елиминисањем строге границе и увођењем одређеног опсега у којем негативни улази не резултирају негативним излазима. Велике позитивне

вриједности су такође проблем јер могу довести до тога да неурон „умре" због нумеричке нестабилности. Како број мртвих неурона расте тако се смањује шанса да процес тренирања исконвергира.

3. Хиперболички тангенс

Хиперболички тангенс је математичка сигмоид функција. Њен облик је:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{27}$$

и приказана је на слици 7в. За разлику од сигмоида, хиперболички тангенс улаз пресликава у опсег $[-1, 1]$.

4. SELU

Скалирана експоненцијална линеарна активациона функција (енгл. *scaled exponential linear unit - SELU*) има математички облик:

$$f(\alpha, x) = \lambda \begin{cases} \alpha(e^x - 1) & x < 0 \\ x & x \geq 0 \end{cases} \tag{28}$$

и приказана је на слици 7г. Параметар скалирања λ је константан док α може варирати. У овом раду усвојене су вриједности 1.0507 за λ и 1.67326324 за α као и у [52]. SELU активациона функција индуковаће самонормализујућа својства само у случају адекватне иницијализације која мора задовољити услов да је њена средња вриједност 0, а варијанса $\sqrt{\frac{1}{N}}$ гдје N представља величину улаза.

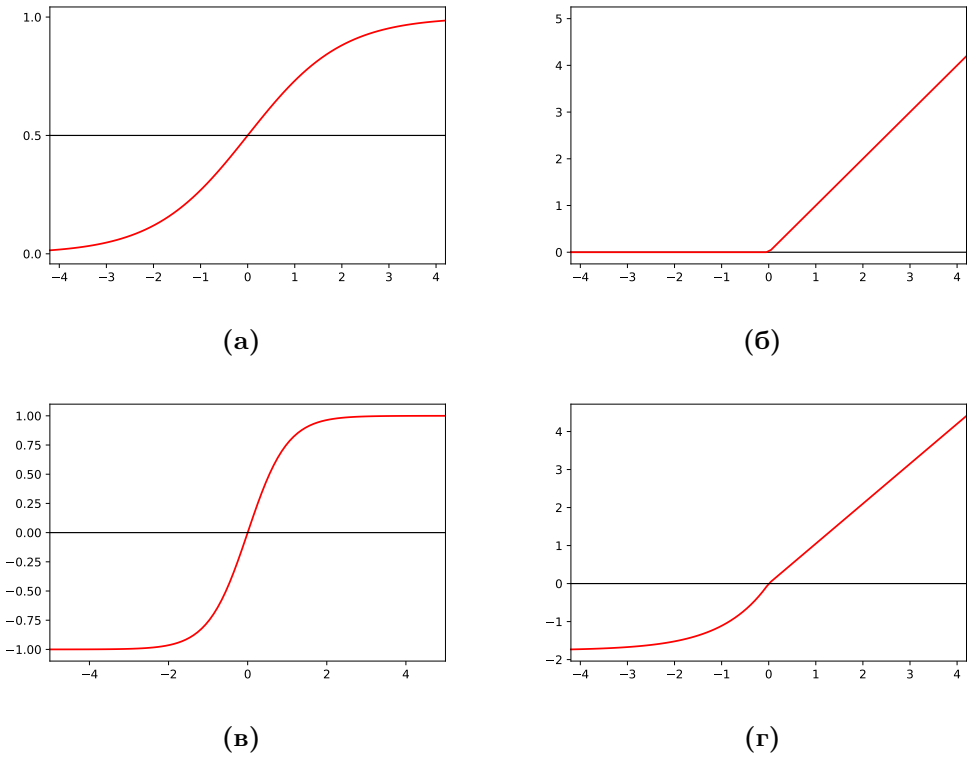
4.2 Конволуционе неуралне мреже

Конволуција се у функционалној анализи дефинише као математичка операција између двије функције x и h чији је резултат функција y која показује на који начин облик једне функције модификује другу. Дефинише се као:

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)\, h(t - \tau)\, d\tau \tag{29}$$

гдје се интеграл може посматрати као бесконачна сума копија сигнала h помјерених за мало временско кашњење и скалиране вриједношћу сигнала x за неки дати тренутак τ. Дискретизована конволуција се дефинише као:

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)\, h(n - k) \tag{30}$$



Слика 7: Активационе функције: (а) SELU, параметар λ је 1.0507 док је α једнак 1.67326324 (б) хиперболички тангенс (в) сигмоид (г) ReLU

Фуријеова трансформација конволуције два сигнала представља производ Фуријеових трансформација тих сигнала. Множење двије Фуријеове трансформације сигнала представља конволуцију у времену. Ово веома важно својство користи се у поглављу 3 за конструкцију Батервортовог филтра као конволуционог слоја неуралне мреже.

У области дигиталне обраде сигнала конволуцијом се дефинише излаз неког линеарног просторно инваријантног система којем је са h означен импулсни одзив система. Линеарно просторно-инваријантни филтри слике представљају 2D конволуцију [53]. Дефинишу се као:

$$y(n_1, n_2) = \sum_{k_1=0}^{K_1} \sum_{k_2=0}^{K_2} h(k_1, k_2)\, x(n_1 - k_1, n_2 - k_2) \tag{31}$$

Уопштено, конволуција се може дефинисати за произвољан број димензија. У неуралним мрежама, конволуциони слој обично представља операцију 2D конволуције тј. прецизније просторне конволуције која се врши над сликама. Конволуциони филтри тј. операција конволуције инспирисала је развијање конволуционих неуралних мрежа (енгл. *Convolutional Neural Network - CNN*) у чијој су основи конволуциони слојеви.

CNN су неуралне мреже које су своје прве успјехе пронашле у компјутерској визији. Поред тога што је њихов основни задатак био рјешавање проблема у дигиталној обради слике, CNN су нашле широку примјену у дубоком учењу. Уопштено, конволуциони слојеви распоређују неуроне у 3 димензије: ширину, висину и дубину која представља број канала. Код 1D конволуције висина је 1 и како се ради о редудантној димензији она се елиминише. За разлику од FC слојева, неурони конволуционих слојева нису повезани са свим претходним неуронима већ локализују сигнал у области која се назива рецептивно поље филтра. Способност локализовања детаља на слици омогућава им боље извлачење особености слике. Како мрежа постаје дубља тако се рецептивно поље слојева повећава јер њихови неурони преко својих улаза виде и улазе претходних слојева. Задатак конволуционог слоја је да следећем слоју преда сажете информације о његовом улазу чиме се филтрирају само есенцијалне особености које мрежа означава као вриједне у рјешавању проблема док се остале занемарују. Што мрежа боље научи да филтрира корисне од бескорисних информација то су њени резултати бољи.

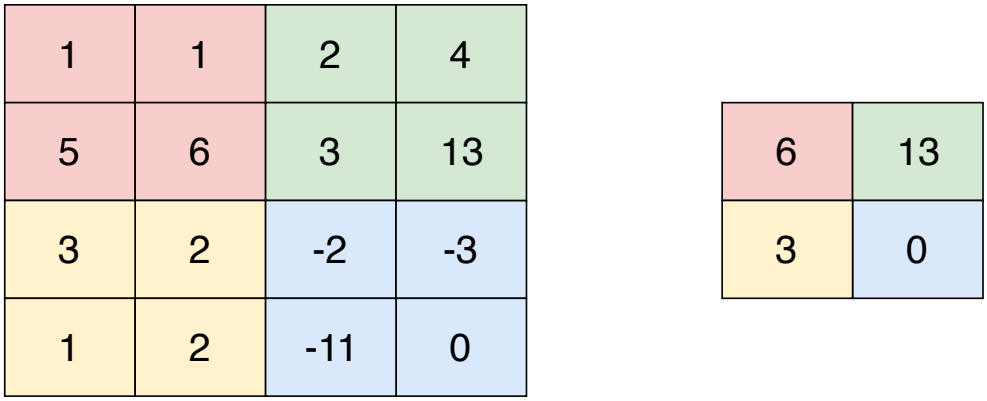
Конволуциони слојеви не утичу на просторне димензије улазних података већ само на број канала. Постоји неколико начина да се просторне димензије улаза смање, а ова операција приказана је на слици 9. Све до недавно у употреби су били слојеви за децимацију од којих је најпопуларнији био такозвани

максимум агрегациони слој (енгл. *max pooling*) којим се од $N \times N$ вриједности, гдје N представља величину филтра овог слоја (обично је N два), бирала само највећа вриједност. Овако агресиван приступ децимацији смањивао је просторне димензије, али и уносио незанемарљиву грешку што се јасно види на слици 8. Због тога је све чешће у употреби смањивање димензија помјерајем којим се неке од вриједности улазног сигнала прескачу. Разлог зашто је грешка код оваквог типа децимације осјетно нижа лежи у малој величини корака у односу на величину филтра. На тај начин ће вриједност која се прескочи ући у прорачун конволуције у случају других вриједности у њеној непосредној околини. На слици 10 приказан је пролазак конволуционог филтра кроз улазну слику употребом помјераја. Како је корак помјераја 2 сваки други пиксел слике се прескаче. Са слике се може уочити да се одређени број пиксела на ивицама неће филтрирати због тога што филтар не може изаћи ван оквира улазне слике. Овај проблем се рјешава додавањем нула по дужини и ширини слике како би се избјегли ивични ефекти. Након конволуирања, излаз слоја пропушта се кроз активациону функцију која се сматра саставним дијелом слоја.

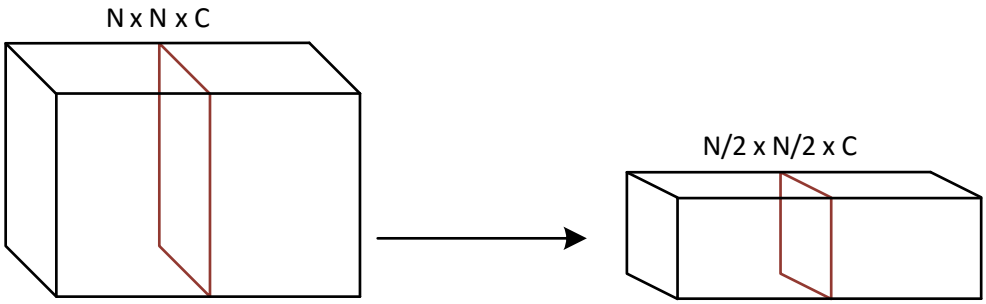
У случају једнодимензионалних сигнала као што је звук, није природно користити вишедимензионалну конволуцију. Код 1D конволуције долази до поједностављења проблема у односу на 2D конволуцију па сада филтар дефинишу само његова дужина и број канала. Рачунање пролаза уназад за конволуциони слој значајно је компликованије од рачунања његовог излаза. Прорачун се може поједноставити наивним приступом без корака помјерања, јединичним бројем канала, без додавања нула и без додавања одступања. Такав приступ је основ за добијање потпуног аналитичког исказа који се може наћи у [54].

4.3 Функција трошка

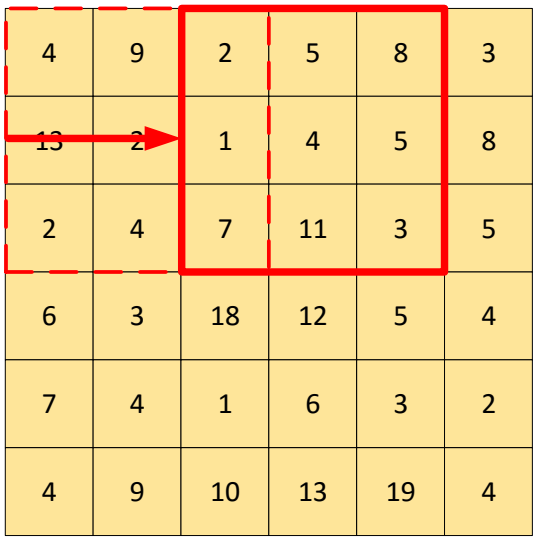
Док алгоритам учи, након сваке итерације врши се провјера његове успјешности над том итерацијом, рачуна функција грешке на основу које се прорачунавају и пропагирају градијенти. Очекује се да ће алгоритам у почетку процеса обучавања правити већи број грешки - алгоритам заправо учи на грешкама. Што је грешка већа то су већи градијенти и обратно. Итерација или серија представља дио корпуса података кроз које алгоритам пролази. Пролазак кроз цијели корпус података назива се епохом. Након обрађивања једне епохе алгоритам се валидира над подацима које до сада није видио, а ови подаци представљају валидациони корпус или његов дио. У случају великог



Слика 8: Примјер агрегационог слоја који децимацију врши помоћу функције максимума. Слој на деструктиван начин одбацује 75% вриједности



Слика 9: Примјер просторне децимације гдје N представља просторну димензију, а C број канала. Врши се по сваком каналу



Слика 10: Филтрирање улазне слике помоћу конволуционог филтра димензија 3×3 и корака помјераја 2

броја података за тренирање, валидација се може вршити и након неколико итерација. Ако би збирку задатака посматрали као податке које ученик треба да пређе како би научио одређену математичку област, једна итерација био би један задатак док би прелажење читаве збирке била епоха. Ученику, а ни алгоритму, једна епоха обично није довољна и кроз податке морају проћи неколико пута. Без обзира на ову алегорију, не може се тврдити да неуралне мреже уче слично људима. То се најбоље може показати на примјеру класификације. Уколико се људском бићу које никад није видјело тигра покуша објаснити шта је тигар, обично ће бити довољно показати му само једну слику тигра на основу које ће у сваком следећем случају то бићe знати да раздвоји тигра од других животиња. Додатно, уколико се човјеку покаже свега неколико слика различитих животиња које припадају породици мачака он ће бити у стању да наредну животињу из ове породице коју до сада није видио успјешно групише. Неуралне мреже имају насилан приступ тренирању - што већи број улазних података то веће шансе да се проблем успјешно савлада. За вријеме тренирања алгоритма неуралних мрежа може доћи до два проблема:

- Преприлагођавање
- Недовољно подударање

Преприлагођавање представља проблем у којем се алгоритам у потпуности прилагоди улазним подацима и изгуби способност генерализације због чега за вријеме тренирања постиже одличне резултате док у валидирању гријеши. До овог проблема може доћи уколико се мрежа предимензионише или уколико се тренирање врши у много већем броју епоха него што је то неопходно. Поједностављивањем мреже или смањивањем броја епоха овај проблем може бити ријешен. Недовољно подударање се јавља онда када се са превише једноставним алгоритмом покуша научити дистрибуција улазних података. Овај проблем се може ријешити повећавањем броја параметара мреже. Сви исходи учења могу се видјети на слици 11.

Функција трошка не може утицати на недостатке мреже, али њен одабир мора одговарати типу проблема који се рјешава. Погрешним избором ове функције умањиће се шансе да алгоритам исконвергира. Пошто се предложени алгоритам састоји из двије неуралне мреже, имаће двије функције трошка. Прва мрежа минимизује грешку реконструкције и као функцију трошка ће користити средњу апсолутну грешку (енгл. *mean absolute error - MAE*). Она се дефинише као:

$$\mathrm{MAE} = \frac{\sum_{n=1}^{N} |y_n - x_n|}{n} \tag{32}$$

где y_n представља предвиђену, а x_n стварну вриједност док је N величина скупа. Друга мрежа је класификатор и користи унакрсну ентропију (енгл. *cross-entropy*). Вјероватноћа да је излаз $y = 1$ се дефинише као:

$$q_{y=1} = \hat{y} = g\left(W \cdot x\right) = \frac{1}{1 + e^{W \cdot x}} \tag{33}$$

где је x улазни податак док је W тежински коефицијент. Вјероватноћа да је излаз $y = 0$ је:

$$q_{y=0} = 1 - \hat{y} \tag{34}$$

Стварне вјероватноће дефинишемо као $p_{y=1} = y$ и $p_{y=0} = 1 - y$. Важи да је $p \in \{y, 1 - y\}$ и $q \in \{\hat{y}, 1 - \hat{y}\}$. Ентропија се дефинише на следећи начин:

$$H\left(p, q\right) = -\sum_{i} p_i \cdot \log\left(q_i\right) = -y \cdot \log\left(\hat{y}\right) - \left(1 - y\right) \cdot \log\left(\hat{y}\right) \tag{35}$$

па је функција трошка:

$$E\left(w\right) = -\frac{1}{N} \sum_{n=1}^{N} \left[y_n \cdot \log\left(\hat{y}_n\right) + \left(1 - y_n\right) \cdot \log\left(1 - \hat{y_n}\right)\right] \tag{36}$$

у којој је:

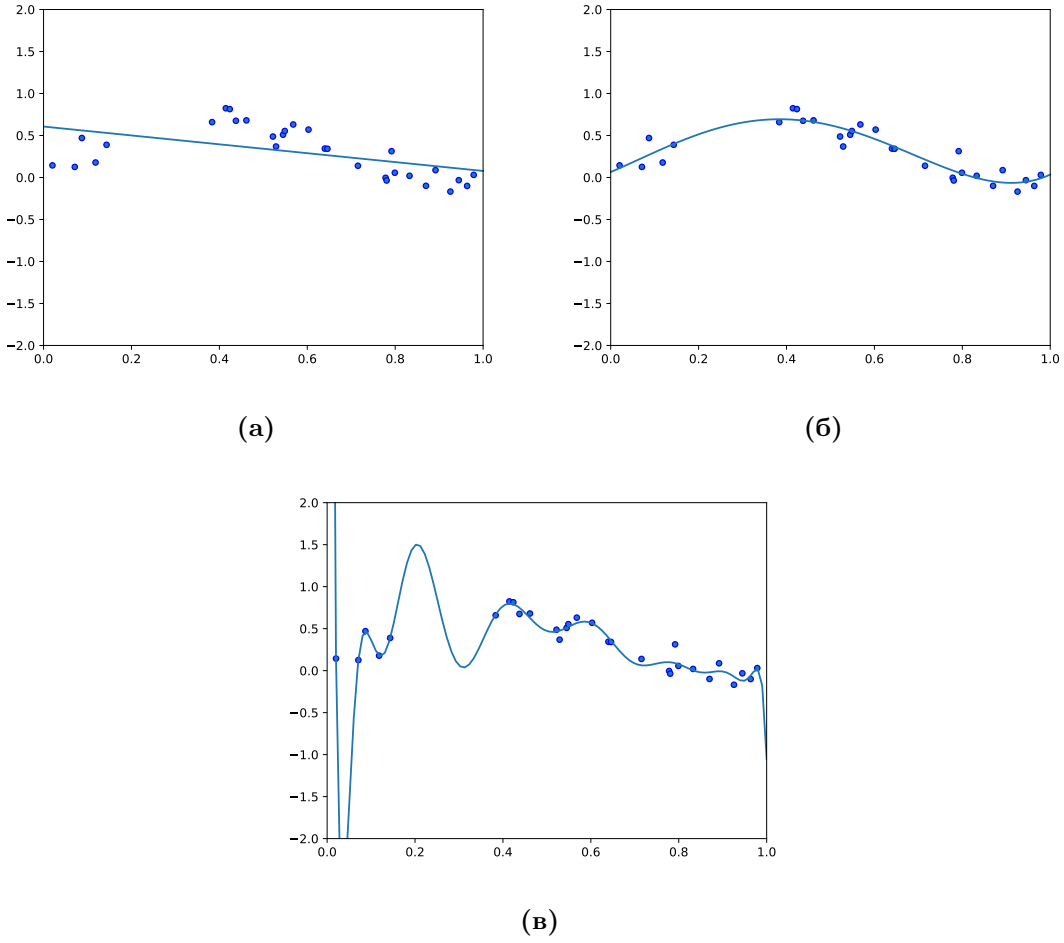
$$\hat{y} \equiv g\left(w \cdot x_n\right) = \frac{1}{1 + e^{-w \cdot x_n}} \tag{37}$$

где $g(z)$ представља сигмоид функцију.

4.4 Алгоритам пропагације уназад

Алгоритам пропагације уназад користи се за минимизацију функције трошка односно за ажурирање параметара мреже. Рачунање се обавља у два корака. Прво се врши пропагација унапријед тј. рачунају се излази сваког слоја од првог до последњег, а затим се рачуна функција трошка након чега се сигнал грешке пропагира уназад чиме се ажурирају параметри сваког слоја. Уколико је излаз неког неурона нула пропагација кроз њега се зауставља.

За дату функцију $f\left(\boldsymbol{x}\right)$ потребно је израчунати извод у односу на вектор $\boldsymbol{x}$. Функција f одговара функцији трошка која се означава са док је $\boldsymbol{x}$ улазна промјенљива. Ову промјенљиву представљају улази слоја, тежински коефицијенти и одступање. У случају првог слоја, улаз представљају подаци над којима се врши обука те се сматрају константним. Парцијални изводи по улазним подацима се рачунају само онда када их је потребно визуелизовати или анализирати. За све остале слојеве улаз је промјенљива и изводи се рачунају



Слика 11: Алгоритам је: (а) недовољно научио (б) добро научио (в) преприлагођен

и по њој јер се пропагирају за слој иза. Тежински коефицијенти и одступања су промјенљиве које алгоритам учи и оне се ажурирају. Треба нагласити да је употребу алгоритма пропагације уназад могуће избјећи, али тиме се не остварује никаква корист. Растом броја слојева број промјенљивих у мрежи драстично расте чиме се коначан израз значајно компликује. Са аспекта практичне имплементације и програмерског рјешења, заобилажење овог алгоритма би такође био лош потез. За показни примјер који слиједи неопходни су нам парцијални изводи функција производа и збира двије промјенљиве као и функције максимума. Рачунањем парцијалних извода [55] функције која представља множење двије промјенљиве:

$$f(x, y) = x \cdot y \tag{38}$$

добијамо:

$$\frac{\partial f}{\partial x} = y, \quad \frac{\partial f}{\partial y} = x \tag{39}$$

Извод представља брзину промјене функције [56] у односу на промјенљиву на бесконачно малом интервалу:

$$\frac{df}{dx} = \lim_{h \to 0} \frac{f\left(x + h\right) - f\left(x\right)}{h} \tag{40}$$

За мале вриједности h функција се у интервалу $\left[f\left(x\right), f\left(x + h\right)\right]$ може апроксимирати правом линијом чији је нагиб једнак вриједности извода. Може се закључити да извод функције по некој промјенљивој представља осјетљивост функције на промјене те промјенљиве. Ако за једначину (38) као примјер узмемо да су $x = 3$ и $y = -2$ тада је $f\left(x, y\right) = -6$. Парцијални извод функције по x износи $\frac{\partial f}{\partial x} = -2$. Повећавањем вриједности x излаз функције смањиће се за два пута више од вриједности за коју се повећала промјенљива x. За дату вриједност y вриједност функције (38) за неко ново x, увећано за h можемо писати као:

$$f\left(x + h\right) = f\left(x\right) + h \cdot \frac{df\left(x\right)}{dx} \tag{41}$$

одакле се јасно види на који начин утиче промјена x. Како извод по y износи $\frac{\partial f}{\partial y} = 3$, повећањем вриједности y за h вриједност функције ће се повећати за $3 \cdot h$.

Градијент ∇ представља вектор парцијалних извода, односно за примјер множења (38):

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right] = \left[y, x\right] \tag{42}$$

и мада је математички нетачно рећи „градијент по x-у", а правилно „парцијални извод по x-у", термин градијент се одомаћио у области неуралних мрежа и користи се као синоним за парцијални извод.

Парцијални изводи функције збира двије промјењиве рачунају се на следећи начин:

$$f(x,y) = x + y \qquad \rightarrow \qquad \frac{\partial f}{\partial x} = 1,\ \frac{\partial f}{\partial y} = 1 \tag{43}$$

а функције максимума:

$$f(x,y) = \max(x,y) \qquad \rightarrow \qquad \frac{\partial f}{\partial x} = \begin{cases} 1, & \text{за} \quad x \geq y \\ 0, & \text{остало} \end{cases}, \quad \frac{\partial f}{\partial y} = \begin{cases} 1, & \text{за} \quad y \geq x \\ 0, & \text{остало} \end{cases} \tag{44}$$

Сада се може увести сложенији примјер функције са три пормјењиве која се састоји из двије операције:

$$f(x,y,z) = (x+y) \cdot z \tag{45}$$

и чије је парцијалне изводе и даље могуће израчунати без алгоритма за пропагацију уназад. Уколико уведемо смјену да је $q = x+y$ добија се да је $f = q \cdot z$. Сада је рачунање парцијалних извода поједностављено. Како је сада f једнака производу функција q и z парцијални изводи су:

$$\frac{\partial f}{\partial q} = z, \quad \frac{\partial f}{\partial z} = q \tag{46}$$

а како је q једнака збиру x и y њени парцијални изводи су:

$$\frac{\partial q}{\partial x} = 1, \quad \frac{\partial q}{\partial y} = 1 \tag{47}$$

Пошто је полазна функција f зависна од x, y и z рачунање парцијалних извода не може се овдје зауставити. По правилу извода сложене функције неопходно је ланчано повезати изразе парцијалних извода смјене q са функцијом f. За промјењиву x по правилу важи да је:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \cdot \frac{\partial q}{\partial x} \tag{48}$$

По овом правилу добија се коначан израз за све парцијалне изводе. Он износи:

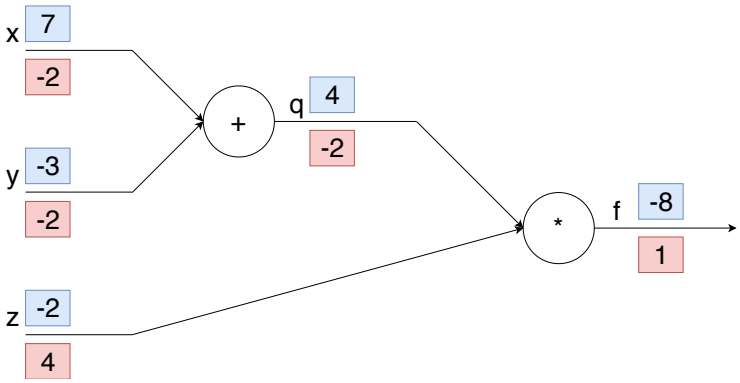
$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right] = [z, z, x+y] \tag{49}$$

Употребом дијаграма кругова над овим примјером може се направити паралела са вјештачким неуронима. Дијаграм кругова је приказан на слици 12. Као што је већ речено, алгоритам пропагације уназад врши рачунање у два корака. Да би се израчунале све вриједности, потребно је проћи кроз дијаграм у оба правца:

1. од лијева ка десно

2. од десна ка лијево

док сагледајући само један круг односно капију могу се израчунати двије вриједности. Једна је излазна вриједност те капије, а друга вриједност локалног извода улаза у односу на излаз. За пропагирање свих извода потребно је сагледати сваку од капија употребом правила извода сложене функције помоћу које се изводи са последње капије преносе до улаза. По овом правилу, вриједност извода који се преноси ка улазу једнак је производу локалног извода капије и оног који је пропагиран од капије десно од ње.



Слика 12: Дијаграм кругова којим је представљено рачунање извода за функцију из примјера (45). Плавом бојом офарбане су вриједности промјенљивих и излаза капија док су црвеном бојом офарбане вриједности извода

Освртом на слику 12 може се уочити да су улази у капију збира једнаки 7 и -3, а излаз односно збир ова два броја је 4. Како се ради о операцији сабирања, оба парцијална извода једнака су 1 по једначини (43). Излаз цијеле функције представљен је капијом множења и износи 8 пошто се на њеном улазу налазе вриједности 4 и -2. Парцијални изводи по q и z једнаки су z и q што је својство операције множења објашњено у (39). Како знамо извод функције по q лако је израчунати изводе по x и y пошто се примјеном правила извода сложене функције зна да се њихов локални извод множи са изводом капије чији су они улази па је $-2 \cdot 1 = -2$. На овом једноставном примјеру који симулира постојање три вјештачка неурона јасно се види предност овог алгоритма. У случају неуралне мреже са бројем од неколико милиона параметара и на десетине или чак стотине слојева алтернатива алгоритму пропагације уназад не постоји.

4.5 Оптимизација

Успјех неуралних мрежа у највећој мјери зависи од доступности и разноврсности података над којим се алгоритам обучава. Како корпус података расте тако се смањују шансе да алгоритам све улазне податке добије истовремено. Подаци се дијеле у серије које се једна за другом пропуштају кроз алгоритам. Трошак се израчунава за сваку серију појединачно. Овај приступ уноси одређену грешку у рачунању извода по параметрима алгоритма, али се не може избјећи због хардверских односно меморијских лимита рачунара. Функција трошка зависи од тежинских коефицијената W. Оптимизација представља процес тражења тежинских коефицијената W таквих да функција трошка има најмању могућу вриједност.

Постоји неколико оптимизационих алгоритама. Основна идеја лежи у праћењу градијента. Наиме, извод функције нам говори у ком правцу и колико стрмо она расте. Ако замислимо функцију трошка као конвексну функцију онда је циљ оптимизације да се што више приближи њеном дну. Извод функције или градијент (за вишедимензионе функције и њене парцијалне изводе) нам говори у ком смјеру да вршимо ажурирање параметара W, али не и за колико да се помјерамо. Стохастички алгоритам градијентног спуста (енгл. *stochastic gradient descent - SGD*) уводи стопу учења (енгл. *learning rate*) помоћу које се врши ажурирање тежинских коефицијената. Представља хиперпараметар мреже и може се мијењати за вријеме тренинга пошто нам генерално одговара да је стопа што већа док смо даљи од циља, а мања како му се приближавамо како га не би прескочили. Ажурирање параметара врши се супротно од смјера градијената зато што је циљ оптимизације да се функција трошка смањи, а не повећа. Како би се ово поглавље поједноставило, компликовани математички изрази биће замијењени Пајтон псеудокодом. SGD алгоритам своди се на:

```
x += - learning_rate * dx
```

гдје је `x` вектор параметара, а `dx` градијент док је `learning_rate` стопа учења.

SGD је основ свих осталих алгоритама оптимизације. Како би се дошло до коначног алгоритма оптимизације који се користи у овом раду неопходно је на кратко увести и оне друге од којих он црпи идеју. Момент ажурирање [57] (енгл. *momentum update*) има физички приступ сагледавања проблема оптимизације. Трошак се може интерпретирати као висина стрмог терена, а самим тим довести и у везу са потенцијалном енергијом $U = mgh$ тј. $U \propto h$. Сада се оптимизација може посматрати као процес симулирања котрљања честице

тј. тежинских коефицијената низ стрми терен. Сила коју ова честица осјећа једнака је градијенту потенцијалне енергије $F = -\Delta U$ односно негативном градијенту функције трошка. За разлику од SGD овај приступ не интегрише на директан начин смјер градијента већ то ради преко убрзања честице. Коначно:

```
v = mu * v - learning_rate * dx
x += v
```

гдје је `v` промјењљива која се иницијализује на 0, а `mu` нови хиперпараметар (обично је 0.9) којим се представља коефицијент трења. Овај оптимизациони алгоритам омогућава да се у смјеру који има конзистентан градијент брзина повећава, а ондје гдје градијент често мијења знак брзина успори чиме се мрежи помаже да исконвергира.

Нестеров момент [58] користи чињеницу да је добар дио следеће итерације (`mu * v`) познат да претпостави колики ће градијент бити у наредном кораку тј. у итерацији `x + mu * v`. Сада се умјесто у $f(x)$, градијент уводи за $f(x + mi \cdot v)$:

```
x_ahead = x + mu * v
v = mu * v - learning_rate * dx_ahead
x += v
```

што је из угла SGD мало неприродно па се уводи смјена $x = x + mi \cdot v$ чиме се увијек чува претпостављена следећа вриједност параметра, а не она тренутна:

```
v_prev = v
v = mu * v - learning_rate * dx
x += -mu * v_prev + (1 + mu) * v
```

Adagrad [59] је оптимизациона метода која уводи адаптивну стопу учења. Имплементира се као:

```
cache += dx**2
x += - learning_rate * dx / (np.sqrt(cache) + eps)
```

Код ове оптимизационе технике се стопа учења дијели са сумом квадратних градијената x до тренутне итерације $\sqrt{s_{xt}}$ (`cache`). Тежински коефицијенти који добијају велике градијенте имаће смањену стопу учења док ће се стопа

учења повећати за оне коефицијенте чији су градијенти мали. Тиме се усклађује брзина ажурирања дуж свих параметара тј. да се SGD процес убрза за оне параметре који се споро уче, а успори за оне који се уче пребрзо. Овим се омогућава да се мрежа што ближе примакне минимуму функције трошка. Вриједност `eps` додаје се због нумеричке стабилности.

RMSprop [60] је ефективан оптимизациони алгоритам који покушава ријешити главни проблем Adagrad технике. Како Adagrad врши неизбјежну акумулацију квадратних градијената, $\sqrt{s_{xt}}$ ће до те мјере порасти да ће се тренирање зауставити. Ово агресивно монотоно опадање стопе учења RMSprop превазилази на следећи начин:

```
cache = decay_rate * cache + (1 - decay_rate) * dx**2
x += - learning_rate * dx / (np.sqrt(cache) + eps)
```

гдје је `decay_rate` нови хиперпараметар и обично узима неку од вриједности $[0.9, 0.99, 0.999]$.

Adam [61] је оптимизациони алгоритам који подсјећа на RMSprop са моментом. Он комбинује предности Adagrad и RMSprop, а име му је изведено из процјене адаптивног момента. Дефинише се као:

```
m = beta1*m + (1-beta1)*dx
mt = m / (1-beta1**t)
v = beta2*v + (1-beta2)*(dx**2)
vt = v / (1-beta2**t)
x += - learning_rate * mt / (np.sqrt(vt) + eps)
```

Хиперпараметри `beta1 = 0.9` и `beta2 = 0.999` контролишу брзину ажурирања параметара алгоритма. За ажурирање се не користи директно градијент `dx` већ његова омекшана верзија `m`. Како се `m` и `v` иницијализују на нулу овај алгоритам имаће проблема у старту док се не загрије. Да би се то избјегло уводи се механизам корекције одступања (енгл. *bias correction mechanism*) тј. нови параметри `vt` и `mt` који зависе од тренутног броја итерације `t`.

Најзад, Nadam [62] или Adam са Нестеровим моментом је унапријеђени Adam алгоритам. Рачуна се на следећи начин:

```
m = beta1*m + (1-beta1)*dx
v = beta2*v + (1-beta2)*(dx**2)
mt = m / (1-beta1**t) + (1-beta1)*dx / (1-beta1**t)
vt = v / (1-beta2**t)
x += - learning_rate * mt / (np.sqrt(vt) + eps)
```

и биће кориштен при обучавању система за уметање воденог жига.

4.6 Нормализација

Дистрибуција улаза једног слоја мијења се за вријеме тренинга како се параметри слојева иза мијењају. Овим се успорава тренирање јер се шанса да се функција трошка заглави у локалном оптимуму повећава. Како би се овај проблем ријешио, слојеви се нормализују. У овом поглављу биће објашњена два приступа у нормализацији. Један од њих је нормализација по серији 4.6.1 која је до недавно била стандард у нормализацији неуралних мрежа. Недавном појавом самонормализујућих неуралних мрежа [52] нормализација по серији добија озбиљну конкуренцију отпорнију на пертурбације података која остварује боље резултате. Како се успјех овог саморегулишућег приступа крије у активационој функцији, она ће бити објашњена у поглављу 4.1 док ће се у овој области пажња посветити слоју за изостављање који се користи у пару са овом активационом функцијом.

4.6.1 Нормализација по серији

Мада је нормализација по серији уведена како би елиминисала ковријантно помјерање улазних података које се јавља током тренинга и повећава како тренирање одмиче она никад није ријешила овај проблем. Оно што је нормализација по серији успјела јесте постизање нулте средње вриједности и јединичне варијансе за слојеве чиме се процес тренирања убрзава и до 14 пута [63]. То се постиже употребом следећег алгоритма:

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \;//\text{ средња вриједност} \tag{50}$$

$$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2 \;//\text{ варијанса} \tag{51}$$

$$\hat{x_i} \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \;//\text{ нормализација} \tag{52}$$

$$y_i \leftarrow \gamma\hat{x_i} + \beta \equiv BN_{\gamma,\beta} \;//\text{ скалирање и помјерање} \tag{53}$$

над серијом. Уведене ознаке $\boldsymbol{\gamma}$ и $\boldsymbol{\beta}$ имају матрични запис јер одговарају карактеристикама улазних података. Пошто се ови параметри уче током тренирања, нормализација по серији се може дефинисати као слој неуралне мреже гдје $\boldsymbol{\gamma}$ представља тежинске коефицијенте, а $\boldsymbol{\beta}$ представља одступање. Нормализација по серији неће бити искориштена у сврху овог истраживања, али је

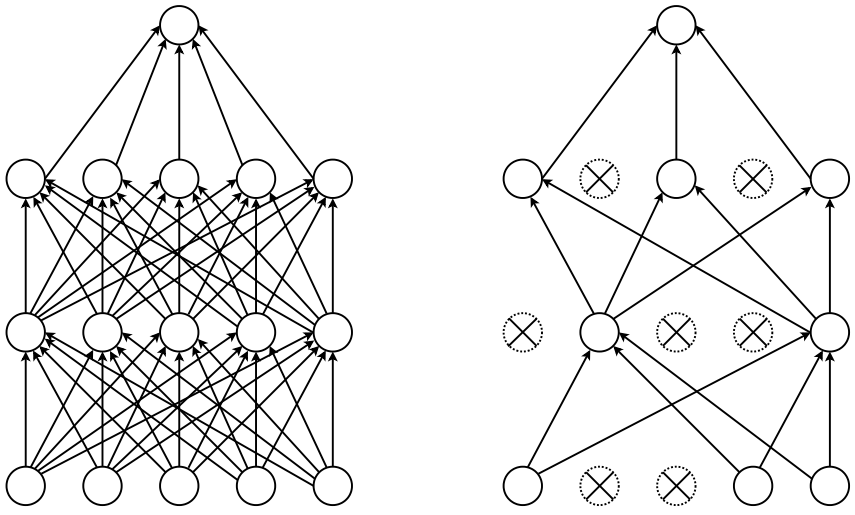
њен теоријски основ важан за разумијевање самонормализујућих мрежа. За вријеме тестирања алгоритам не ажурира параметре слојева већ се користи онима који су већ израчунати.

4.6.2 Алфа изостављање

Изостављање представља једноставну регулизациону технику помоћу које се смањује шанса за преприлагођавањем [64]. Врши се тако што се насумичан број неурона неког слоја у пролазу ка напријед поставља на нулу. Ова операција приказана је на слици 13. Стандардно изостављање поставља неуроне на 0 са вјероватноћом $1 - q$ гдје је $0 < q \leqslant 1$. Како би се очувала средња вриједност за вријеме тренирања излаз слоја се множи са $1/q$. ReLU активациона функција се уклапа добро са изостављањем јер је нула у региону ниске варијансе и одговара подразумијеваној вриједности. Међутим, за SELU активациону функцију, подразумијевана вриједност у региону ниске варијансе је:

$$\lim_{x \to -\infty} \operatorname{selu}(x) = -\lambda\alpha = \alpha' \tag{54}$$

те се у случају SELU активационе функције уводи појам алфа изостављања код којег се изостављени неурони постављају на вриједност α'. Аутори алфа изостављања наводе да је вриједност q најбоље изабрати тако да је $1-q = 0.05$ или $1 - q = 0.10$.



Слика 13: Примјер стандардне неуралне мреже са 2 скривена слоја (лијево) и приказ исте те мреже након примјене изостављања гдје су прекрижени неурони они неурони који су изостављени

5 Предложена архитектура

Архитектура DNN представљена у овом раду дјелимично се заснива на [41], може се сматрати наставком истраживања [65] и приказана је на 17. Састоји се од двије цјелине, уметача и детектора. Мреже уметача и декодера описане су у поглављима 5.1 и 5.2. DNN се обучава да савлада нападе описане у поглављу 3. Као што је већ речено, напади се понашају као интегрални дио мреже. Њихови параметри су константни и не могу се мијењати током тренирања, али њихов положај у мрежи чини их дијелом DNN јер градијент пролази кроз њих од декодера до уметача. Како њихово присуство уноси константан проток градијената јер се њихови параметри не мијењају, неће се посматрати као независни дио мреже. Кратак преглед читаве архитектуре може се видјети у табели 1. Укупан број параметара обје мреже је 4.346.025 што ову архитектуру чини умјерено комплексном. Мрежа за нападе има свега 400 параметара који се не обучавају. Оптимизација алгоритма врши се помоћу Nadam оптимизационе технике, објашњене у поглављу 4.5.

Мрежа	Димензије улаза	Димензије излаза	Број параметара
Уметач	32768×1	32768×1	2948425
Напади	32768×1	32768×1	400
Детектор	32768×1	128×1	1397600

Табела 1: Табеларни приказ димензија улаза и излаза уметача и детектора и њихов број параметара

5.1 Уметач

Неуралне мрежа уметача заснива се на U-net дизајн концепту [66] који је првобитно кориштен за сегментацију слика, али је нашао широку намјену. Ова мрежа је структурно веома слична аутоенкодерима. Шема основног аутоенкодера се може видјети на слици 14. Аутоенкодери су неуралне мреже које трансформишу улаз x у неку латентну просторну репрезентацију r, након чега ту просторно компресовану репрезентацију експандују до $\hat{x}$. У идеалном случају аутоенкодери теже да задовоље:

$$x = \hat{x}. \tag{55}$$

Састоје се из енкодера и декодера. Задатак енкодера је да компресује сигнал до r, а декодера да га реконструише до $\hat{x}$. На први поглед не дјелује корисно што

аутоенкодер покушава да научи преносну функцију $f(x) = x$, али вриједност аутоенкодера налази се у пресликавању улаза у латентни простор r.

Аутоенкодер представља облик ненадгледаног учења без обзира што је жељени излаз заправо улаз мреже јер он нема јасне лабеле. За потребе овог истраживања користиће се конволуциони аутоенкодер у којем се умјесто FC слојева користе конволуциони слојеви који се обично користе у обради слике. У случају звука који представља једнодимензионалне податке, конволуциони слојеви аутоенкодера за звук морају користити 1D конволуцију тј. конволуциони неуралнi слој који се дефинише дужином прозора и бројем филтара.

У случају U-net дизајн концепта, енкодер и декодер мреже су симетричне. Аутоенкодери се углавном користе у екстракцији карактеристика сигнала, али и онда када је потребно наћи просторно компактнију репрезентацију неког сигнала. За потребе овог истраживања, аутоенкодер се користи како би систем пронашао вјештачки трансформациони домен у који би могао да смјести односно да сакрије водени жиг. Уметање воденог жига у репрезентацији r отежава стандардни задатак аутоенкодера чиме се дјелимично нарушава његов концепт. Међутим, основни задатак из једначине (55) остаје непромијењен. Аутоенкодер мора настојати да у потпуности елиминише водени жиг, односно да смањи разлику између x и $\hat{x}$. Како је циљ система да водени жиг преживи уметање и нађе се у сигналу носиоцу постаје јасно да се процес тренирања не може посматрати засебно у односу на мреже уметача и детектора. Уметач ће настојати да елиминише водени жиг, али ће градијент који се спушта из детектора кажњавати уметач уколико се примакне том циљу чиме ће се увијек остављати простор оптималне конвергенције за обје мреже. Трансформациони домен који генерише уметач зависи од улазних података, али и од самог начина тренирања као и од иницијализације његових параметара. Пошто градијенти зависе од улазних података, промјена параметара уметача зависиће у највећој мјери од врсте података над којима се мрежа тренира. У случају затворених система уске примјене ово је са аспекта безбједности веома добро, али у случају опште генерализације одређена доза сумње је оправдана. Водени жиг представљен је као једнодимензионална бинарна порука фиксне величине. Порука може имати неко значење, али је за потребе овог истраживања генерисан скуп случајних порука. Умјесто насумичне секвенце битова могла се генерисати ASCII порука или чак бинарна слика. Подаци се током тренинга насумично мијешају што је добра пракса код обучавања неуралних мрежа. Мијешањем података и случајном иницијализацијом практично се онемогућава обрнуто инжењерство чак и онда када би нападачи добили приступ корпусу података. Архитектура уметача приказана је на слици 16. Улаз уме-

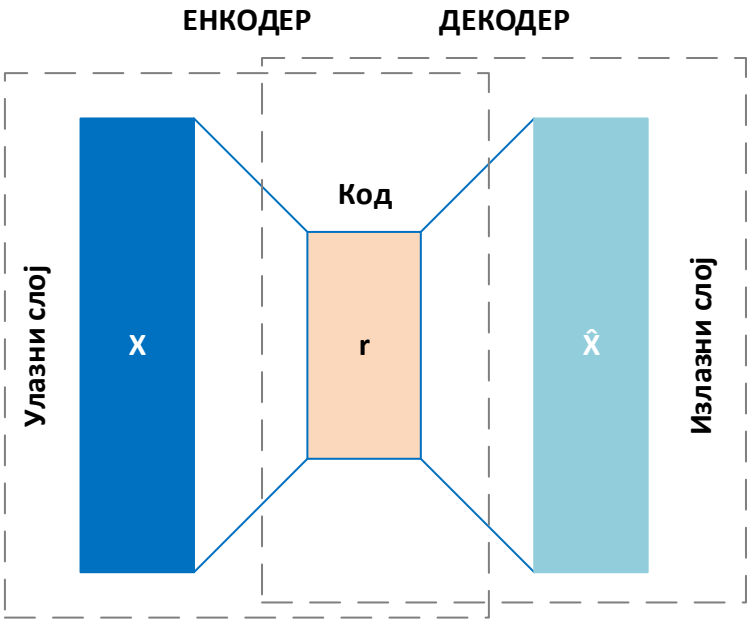
тача представљају одбрици аудио сигнала у временском домену. Подаци које уметач добија на улазу су димензија 32768×1 гдје 32768 представља дужину сегмента који се предаје уметачу. Коришћење сирових података је са тачке гледишта DNN више природно јер избјегава предобраду података и процес тренирања чини једноставнијим елиминисањем процеса одабира трансформационог домена.

Први дио мреже уметача састоји се од 6 блокова и врши децимацију (смањено узорковање) улазних података. Ови блокови сачињени су од 1D конволуционог слоја са помјерајем 1 за први блок, 8 за други, 4 за трећи и 2 за остале. Величина језгра (кернела) односно прозора којим се врши конволуција је 41 за први блок, а 21 за остале блокове. Број филтара је дијадичан односно 2^k гдје је $k \in [3..8]$. Употреба једнодимензионалне конволуције условљена је природом улазних података. Како је задатак децимационог блока да редукује просторне димензије сигнала користе се велики помјераји и још већи прозори. Овако агресивне величине прозора и помјераја могу се оправдати. Сигнал око тачке t у времену може се прогласити константним ако је временски офсет Δ довољно мали. То значи да се у интервалу $[t - \Delta, t + \Delta]$ може сматрати да је сигнал константан. Како је фреквенција узорковања 16000, за једну милисекунду имамо 16 одбирака. Може се очекивати да се сигнал неће значајно промијенити за нешто више од двије милисекунде. Примјер се може видјети на слици 15. Ова апроксимација није у потпуности тачна, али упарена са помјерајем који је значајно мањи од прозора чиме сваки одбирак бива филтриран по неколико пута, може се усвојити као вјеродостојна. Дакле, употреба језгра величине 41 је оправдана. Величина језгра смањује се како се смањују просторне димензије улаза. Смањивањем просторних димензија повећава се и рецептивно поље наредних филтара односно што се више просторне димензије смањују тако филтри посматрају већи број улазних одбирака. Када се величина филтра не би смањила постојала би могућност да се одређени број информација изгуби. За нормализацију података не користи се нормализација по серији већ SELU која сама индукује нормализујућа својства [52]. За иницијализацију слоја који користи SELU активациону функцију мора се користити LeCun иницијализација [52, 67] како би се искористила нормализујућа својства која пружа SELU. Ова активациона функција је дефинисана у поглављу 4.1.

Резултујућа репрезентација сигнала има димензије 128×256. Премда је просторна димензија драстично смањена, број канала је веома велики те се меморијска захтјевност није смањила. Критички сагледано, димензије сигнала су остале исте јер је $128 \cdot 256 = 32768$, али циљ ове мреже није ни био да се

димензионалност улазног сигнала смањи већ да се пронађе трансформациони домен у који је могуће уметнути водени жиг. Смањивањем димензија сигнала смањиле би се и шансе за потпуну реконструкцију што би се негативно одразило на цијели систем. Водени жиг се додаје тако што се надовезује тј. спаја на крају димензије за канале што се разликује у односу на начин који је предложен у поглављу 2. Како је водени жиг једнодимензионални низ, спајање не може се извршити директно због очигледних димензионих неслагања. Репрезентација улазног сигнала у латентном простору сада има двије димензије и може се посматрати као правоугаоник. Водени жиг се може посматрати као дуж. Пошто се ради о дискретним вриједностима, висина воденог жига је један. Посматрано тако, репрезентација улазног сигнала може се замислити као 128 дужи дужине 256 наслаганих једна на другу. Да би се то тијело спојило са воденим жигом, висина воденог жига се мора повећати 128 пута. То се може урадити на два начина, уметањем нула или његовим понављањем. Уметање нула је валидно рјешење, али за овај случај резултат би био незадовољавајући. Како би се повећале шансе да се водени жиг очува до излаза уметача, водени жиг се понавља и тек онда додаје репрезентацији улазног сигнала. Резултат спајања нарушава симетрију U-net концепта чиме се утиче на други дио ове мреже који врши интерполацију. Овај проблем се рјешава увођењем још једног конволуционог слоја који има 256 филтара, помјерај 1 и језгро величине 9. Поново се користи SELU активациона функција те LeCun иницијализација.

Други дио мреже уметача се такође састоји од 6 блокова и врши интерполацију улазних података. Прва три блока сачињени су од 1D транспонованих конволуционих слојева са помјерајем 2 и величином језгра 21. Као активациона функција користи се SELU, а слојеви се иницијализују помоћу LeCun иницијализације. Блокови се такође састоје и од алфа изостављања објашњеног у поглављу 4.6.2 чији је проценат изостављања 40%. Слој алфа изостављања уведен је заједно са SELU активационом функцијом у [52] те не нарушава њену способност нормализације. Нормализација по серији је у потпуности изостављена. Употреба изостављања је наиван приступ покушаја остваривања робустности, а и заштита од претренирања. Изостављањем великог дијела излаза одређеног слоја мрежа бива присиљена да се прилагоди нападима. Препорука из [52] је да се користи значајно већи проценат изостављања, али како је задатак мреже да при реконструкцији направи што мање пропуста овај проценат је смањен чиме се постиже компромис. Следећа два блока дијела за интерполацију имају помјерај 4 односно 8 док је њихова величина језгра 21, а алфа изостављање се не користи. Излази одговарајућих блокова за деци-



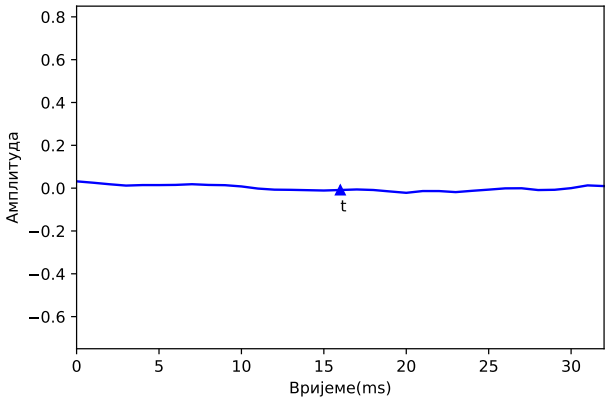
Слика 14: Аутоенкодер

мацију спајају се са одговарајућим излазима блокова за интерполацију чиме се олакшава уметачу да реконструише сигнал. Транспоновани конволуциони слојеви имају 2^k филтара гдје је $k \in [7..3]$. Како се димензије улаза и излаза не подударају уводи се конволуциони слој чије је језгро дужине 41, помјерај је 1, а број филтара 1. Овим се број филтара смањује са 8 на 1 чиме се димензије изједначавају. Функција трошка коју уметач користи је средња апсолутна грешка.

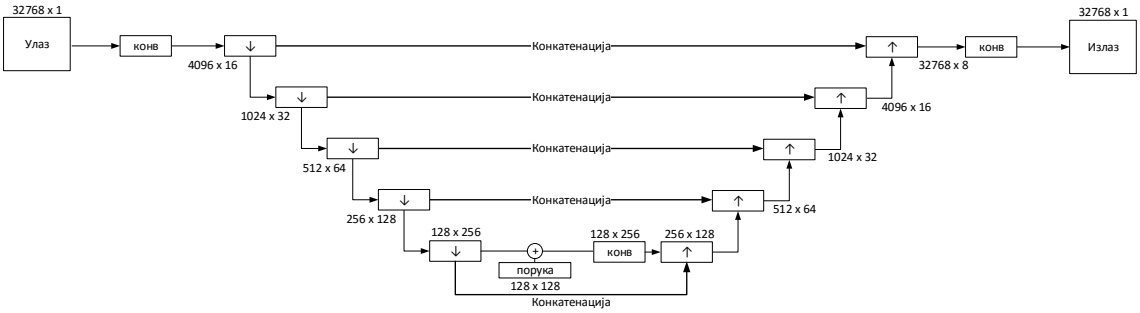
Вриједности улазног аудио сигнала су ограничене. Зависно од начина записивања оне могу бити позитивне и негативне. Како SELU активациона функција није симетрична у односу на y осу, а ни не ограничава амплитуду излаза она не може се користити на последњем слоју јер не може на добар начин моделовати излаз који је по својој природи аудио запис. Због тога се као активациона функција користи хиперболички тангенс описан у поглављу 4.1, а за иницијализацију се користи Xavier односно Glorot иницијализација [68]. Након излаза из уметача, реконструисани сигнал пролази кроз серију напада. Пошто су ови напади дефинисани у временском домену над сигналом није потребно вршити никакву трансформацију.

5.2 Детектор

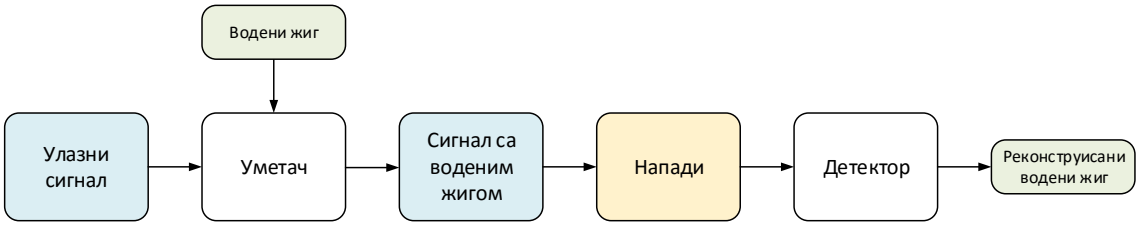
Архитектура детектора приказана је на слици 18. Улаз детектора јесте реконструисана верзија улазног сигнала односно излаз уметача. Тачније, улаз



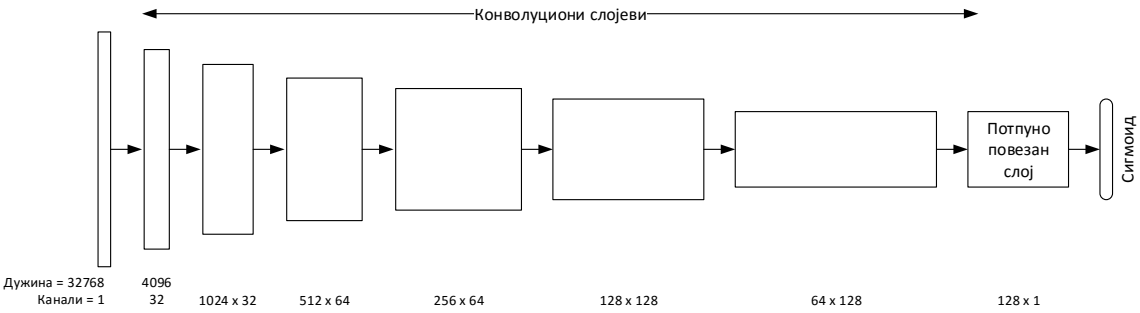
Слика 15: Сигнал се може сматрати константим у интервалу $[t - \Delta, t + \Delta]$ за довољно мало Δ



Слика 16: Архитектура уметача. Блокови за децимацију (↓) су конволуциони слојеви који смањују просторне димензије. Блокови за интерполацију (↑) су транспоновани конволуциони слојеви који повећавају просторне димензије.



Слика 17: Архитектура



Слика 18: Архитектура детектора. Први слој представља улазне податке.

детектора јесте реконструисана верзија сигнала која је потенцијално прошла кроз серију напада. Како сигнал бива нападнут са одређеном вјероватноћом, декодер може добити и сигнале који су ненападнути. Као што је већ речено, детектор је дизајниран по угледу на неуралне мреже за класификацију - класификаторе. Логички сагледано, детектор није класификатор, али са становишта DNN он се може сматрати таквим. Његов задатак је да класификује сваки бит воденог жига понаособ и утврди да ли се ради о 0 и 1. Након тога детектор одлучује о којој поруци је ријеч пошто је број порука ограничен и износи 8. Функција трошка коју детектор користи је бинарна унакрсна ентропија. Она се рачуна између излаза детектора и воденог жига уметнутог у уметачу тј. оригиналне поруке.

Детектор се састоји од 6 децимационих блокова. Блокови су сачињени од 1D конволуционих слојева чија су језгра димензија 41 за први, 21 за други и трећи, 11 за четврти и пети и 9 за последњи, шести блок. Помјерај у првом блоку је 8, другом 4, а у осталима 2. Број филтара прва два слоја је 32, друга два 64 и последња два 128. Конволуциони слојеви имају SELU активациону функцију, а иницијализују се уз помоћ LeCun иницијализације. Излаз из последњег конволуционог слоја трансформише се у једнодимензионални сигнал, а након тога пролази кроз FC слој. Број активација FC слоја једнак је дужини воденог жига, а овај слој користи Glorot иницијализацију. Активациона функција потпуно повезаног слоја је сигмоид, дата у поглављу 4.1. Пошто је неопходно да излаз детектора има димензије воденог жига број блокова и њихови помјераји као и број филтара се прилагођавају овом захтјеву.

6 Тренирање

Како се систем садржи од двије неуралне мреже са посебним функцијама трошка, процес тренирања је нешто компликованији у односу на системе са једном неуралном мрежом. Обуку додатно отежава чињеница да су ове двије мреже супростављене једна другој и да је успјех једне заправо неуспјех друге. Задатак система је да пронађе оптимално рјешење у којем обје мреже имају задовољавајући резултат. У случају да једна мрежа надвлада другу угрозиће њену способност учења чиме ће читав систем доживјети неуспјех. Улаз детектора је сигнал у којем се налази водени жиг, то је предуслов за његов успјешан рад. Уколико се у сигналу не налази водени жиг, детектор не може учити да врши класификацију. Како информација коју сигнал носи детектору није важна, његов основни задатак у почетним фазама тренирања јесте да осигура долазак воденог жига на његов улаз. То је могуће зато што су детектор и уметач повезани и градијенти се од функције трошка детектора сливају све до уметача. Уметач настоји да изврши потпуну реконструкцију сигнала, а водени жиг несумњиво уноси непрецизност због чега ће уметач настојати да га обрише. Због тога је неопходно на контролисан начин осујетити уметач за вријеме тренирања, тако да се детектору не да превелика предност чиме би се угрозио квалитет реконструкције, процеса једнако важног колико и детекција воденог жига.

Како је на процес тренирања неопходно утицати, трошак детектора и уметача скалира се помоћу тежинских фактора чиме се осигурава конвергенција обје неуралне мреже. Тежински фактори су функција тренутног броја епохе и дефинишу се као:

$$w_e(t) = \begin{cases} \Pi_{w_e} & t \leq 1 \\ \Pi_{w_e} + (t-1) \cdot \kappa_{w_e} & 1 < t \leq K \\ \Phi_{w_e} & K < t \leq N, \end{cases} \tag{56}$$

$$w_d(t) = \begin{cases} \Pi_{w_d} & t \leq 1 \\ \Pi_{w_d} - (t-1) \cdot \kappa_{w_d} & 1 < t \leq K \\ \Phi_{w_d} & K < t \leq N \end{cases} \tag{57}$$

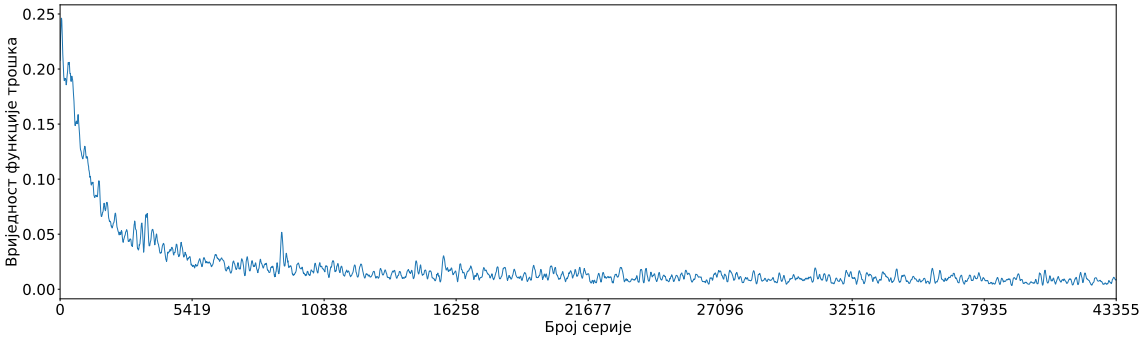
гдје су $w_e(t)$ и $w_d(t)$ вриједности тежинских фактора уметача и детектора и функција су од t, тренутног броја епохе. Величине Π_{w_e}, Φ_{w_e}, κ_{w_e}, Π_{w_d}, Φ_{w_d}, κ_{w_d}, K и N представљају параметре помоћу којих се одређују вриједности тежинских фактора. Са Π и Φ означене су почетна и финална вриједност док је κ корак. Параметри K и N су произвољне епохе. Вриједности ових параметара које су коришћене при тренирању алгоритма дате су у табели 2.

Одређене вриједности, као N могу се сматрати редудантним, али су остављене због генерализације рјешења - како се ради о хиперпараметрима мреже они могу узети произвољне вриједности.

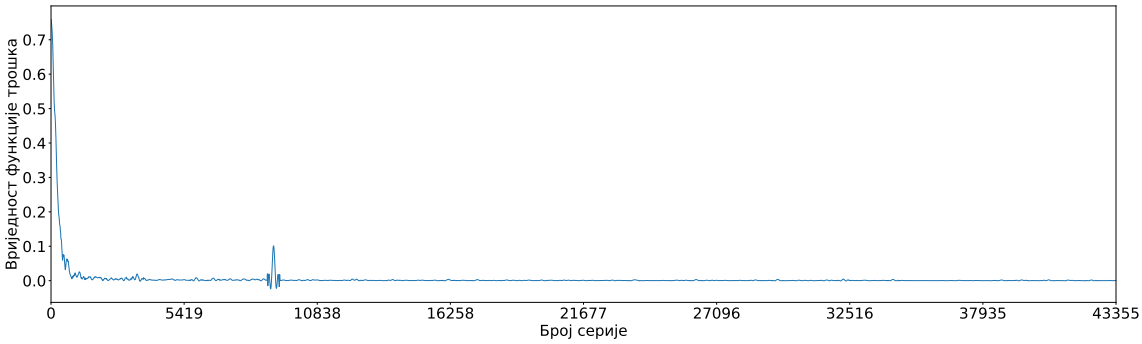
Одредити ове хиперпараметре аналитичким путем није лако. Ради се о комплексном проблему са више промјењљивих који зависи првенствено од величине корпуса података. Постоје одређене олакшице, тренирање се може прекинути одмах чим буде јасно да функција трошка не опада и да мрежа дивергира, али у великом броју случајева резултати ће бити задовољавајући што нужно не значи и најбољи. Пошто је вријеме тренирања и валидације јако велико па чак и у случају употребе најсавременијих графичких картица, аналитичко одређивање није имплементирано у склопу овог истраживања. Умјесто тога, хиперпараметри су одређени емпиријски. У току почетне фазе тренирања уметач се много боље сналази од детектора па се подешавањем већих тежинских фактора за њега покушава спријечити рани неуспјех. Тежински фактор декодера се постепено смањује док се тежински фактор увећава како би се грешка реконструкције минимизовала. Изабрати добре почетне вриједности је веома битно. Превелик тежински фактор довешће до нестабилности док ће премали онемогућити мрежи да научи.

Одређен број серија које су улаз декодера бивају нападнуте како би систем научио да се брани од напада на водени жиг који су презентовани у поглављу 3. Слојеви неуралне мреже који врше нападе имају хиперпараметар који представља вјероватноћу да ће доћи до тог напада. Како је одређивање овог параметра ван опсега овог истраживања одређено је да тај параметар има вриједност 15%.

Вриједности функције трошка уметача и детектора приказане су на сликама 19 и 20 као функција од броја серије. Да би се графици боље читали вриједности функције трошка филтрирани су помоћу Савицки–Голај филтра [69] са прозором величине 51 и полиномом реда 3. Јасно се уочава способност детектора да за нешто више од једне епохе исконвергира и спусти вриједност функције трошка на нулу док се мрежа уметача мучи, има огромне осцилације трошка, али се тренд пада ове вриједности може уочити све до самог краја тренирања. Обучавање се врши у свега три епохе и прекида. Очигледно је да готово и да не постоји шанса да ће уметач и даље наставити да учи уколико се сагледа само његов трошак, али други параметри којима се мјери квалитет реконструкције показују константан помак те се обучавање може продужити. Међутим, како се овај помак при крају тренирања значајно смањује, осјетно бољи резултати се не могу остварити.



Слика 19: Вриједност функције трошка уметача

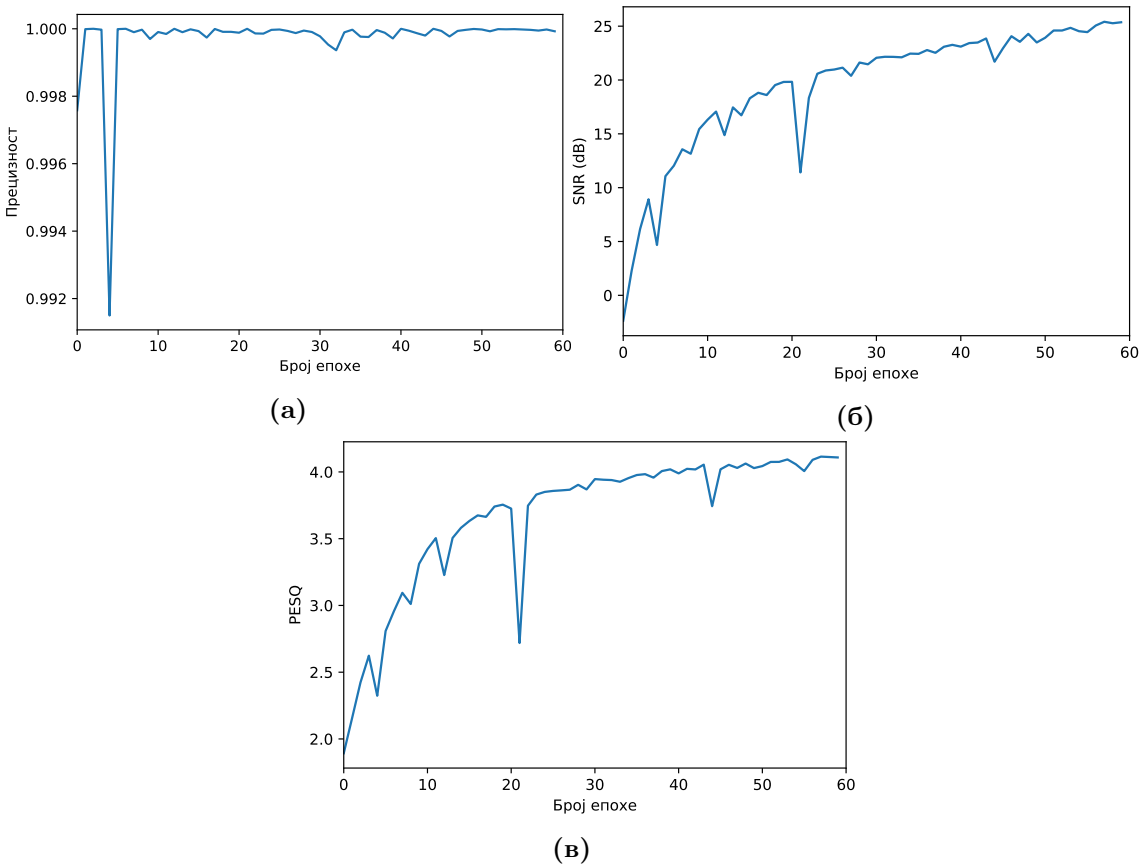


Слика 20: Вриједност функције трошка детектора

Треба напоменути да је број епоха „лажан”. Корпус података је огроман и интервенција над тежинским факторима мора се извршити много прије краја једне епохе којих је 3, а не 60. Како је незахвално користити појам двадесетина епохе уведена је смјена 1 епоха једнако 20 „лажних” епоха.

Параметар	Вриједност
Π_{w_e}	1
Φ_{w_e}	3
κ_{w_e}	0.2
Π_{w_d}	2
Φ_{w_d}	1
κ_{w_d}	0.1
K	10
N	10

Табела 2: Вриједности параметара алгоритма помоћу којих се израчунавају тежински фактори



Слика 21: Тренд метода за мјерење перформанси: (a) Прецизност детекције воденог жига, (b) SNR сигнала у који је уметнут водени жиг и (c) PESQ сигнала у који је уметнут водени жиг

7 Корпус података

Успјех алгоритама дубоког учења зависи од корпуса података над којим се алгоритам обучава. Може се тврдити да је све већа приступачност подацима директно позитивно утицала на резултате постојећих рјешења [70, 71]. Алгоритми дубоког учења биће у стању да уче брже и побољшају своје перформансе када на свом улазу имају свеобухватне и на квалитетан начин припремљене податке. Овај неупитан успјех не може се наравно приписати само величини корпуса већ и општем квалитету одређеног алгоритма, али и дубини мрежа дубоког учења. Како би се успјех једног алгоритма поредио са другим алгоритмима неопходно је обезбиједити идентичне предуслове. Општи предуслов јесте референтни корпус података. Да би се убрзао развој дубоког учења и ријешили одређени проблеми, захваљујући јавно-приватном партнерству уложен је огроман труд на креирању огромних референтних корпуса података који су јавно доступни. Са преко 14 милиона слика, ImageNet [72] је одличан примјер успјешности референтних корпуса података јер је уз његову помоћ ријешено неколико проблема из области рачунарског вида од којих је класификација слика најзначајнији.

Област класификације звукова доживјела је ограничен успјех појавом референтих корпуса од којих је један ESC [73]. За разлику од рачунарског вида и многих других области, још није усвојен стандардизовани референтни скуп података за оцјену квалитета система за уметање воденог жига у области аудио сигнала, а посебно не у области говора. Стога је тешко објективно упоредити различите алгоритме који се баве овом проблематиком. У случају поређења алгоритама дубоког учења са традиционалним методама овај проблем бива израженији. Тренутно су у употреби бројни аудио корпуси података. Аутори [33] користе TIMIT [74] корпус података док се SQAM [75] корпус користи у [39].

Направити генерализовани систем који врши уметање воденог жига у аудио сигнале било којег типа је са аспекта дубоких мрежа јако тежак задатак. Они се могу разликовати по фреквенцији одабирања, висини тона, дужини, ... Ово истраживање бави се искључиво уметањем воденог жига у дигиталне записе људског говора. Посебан проблем који овај систем за уметање воденог жига покушава да ријеши јесте заштита изјава политичара и других јавних личности, али и да потврди аутентичност извора аудио записа. За потребе тренирања обезбијеђен је скуп аудио записа сачињен од стране Скупштине Црне Горе чија је законска обавеза да снима све сједнице. На снимцима говоре посланици и бројни гости Скупштине. Мада се ради о јавним информацијама

њих тренутно није могуће преузети без одобрења. Постоји нада да ће се од ових записа направити први референтни корпус података за потребе поређења радова на тему уметања воденог жига у говорне сигнале.

Од скупа аудио записа из Скупштине Црне Горе направљен је корпус података кориштен за тренирање и валидацију алгоритма. Корпус се састоји од 6199 снимака дужине од 10 минута који су прикупљени за вријеме 26. сазива и то у периоду од 2016. па све до 2019. године. Најмлађи говорник имао је 22 године, а најстарији 73. Било је 280 говорника од којих су неки говорили свега неколико секунди док су најактивнији говорили од 3 до 4 часа. Мултикултуралност Црне Горе утицала је на разноврсност дијалеката и нагласака који се поред многих других карактеристика говора као што је динамика, звучност и сл. јављају у овом корпусу. Због малог броја жена у политици Црне Горе (испод 30%) присутан је очигледан дисбаланс у полу говорника. Фреквенција одабирања сигнала је 44.1 KHz што је и више него довољно за говор. Одређен број снимака упитног је квалитета због повремене појаве шума, буке али и вербалних сукоба који су чести у Скупштини Црне Горе. Број таквих снимака је занемарљив и не утиче битно на квалитет корпуса.

Различитости у гласовима и што већи број говорника позитивно утичу на рад алгоритма чиме се поспјешује његова способност да генерализује, а смањује могућност преприлагођавања. Огроман корпус података неће увијек бити од користи уколико су снимци незадовољавајућег квалитета, а разноврсност података недовољна. Људски слух је углавном ограничен на опсег од 20 Hz до 20 kHz, а тај опсег се сужава како старимо. Фреквенција нашег гласа ријетко прелази 8 kHz [76]. У великом броју случајева крајња граница је у опсегу од 1 до 5 kHz [77]. Ови научни закључци су у доброј мјери утицали на ширину опсега ускопојасних телефонских система која износи свега 3.4 kHz. Овако ниска фреквенција ће несумњиво утицати на квалитет гласа који се преноси, али не и на информацију. Пошто предложени систем врши обраду говорних сигнала, а како је фреквенција одабирања на нивоу читавог корпуса 44.1 kHz она се може значајно смањити што ће унијети занемарљив утицај на квалитет сигнала. Децимација фреквенције одабирања може утицати на квалитет сигнала, али величина улазних података неуралне мреже драстично утиче на њене перформансе. Смањивањем димензија улазних сигнала смањују се хардверски захтјеви и убрзава се рад мреже. По Никвист-Шеноновој теореми одабирања [78], фреквенција одабирања мора бити за макар два пута већа од максималне фреквенције сигнала како би се омогућила идеална реконструкција континуираног сигнала из његове дискретне репрезентације. Како је жељена максимална фреквенција 8 kHz за фреквенцију одабирања

одабрана је вриједност од 16 kHz.

Говорници у својим излагањима могу правити паузе, али исто тако се за вријеме сједнице могу десити ситуације у којима је свим говорницима искључен микрофон. Примјећено је да ове паузе могу трајати и по неколико минута. Пошто паузе не представљају корисне информације у интересу је да се оне одстране. Ово је веома важно и због успјеха неуралне мреже - њен задатак није да умеће водени жиг у шум, а још мање да учи реконструкцију шума који представља напад на систем. Сви интервали дужи од 1 s у којима није забиљежен говор су уклоњени. Експериментално је утврђено да је праг испод којег се све сматра тишином око -35 dBFS. Операција уклањања тишине је корпус података смањила за чак 165 сати па је укупна расположива дужина 868 сати.

Након уклањања тишине снимци су подијељени у сегменте дужине 32768 што представља нешто више од 2 секунде. Последњи сегмент у секвенци допуњен је нулама уколико му је дужина мања од очекиване. Овај приступ има неколико недостатака, али значајно олакшава претпроцесирање података. У најгорем случају гдје сваки последњи сегмент има свега један одбирак у корпус се умеће 206 минута тишине, али овај случај је мало вјероватан. Како то нуле утичу на рад алгоритма? Пошто се конволуција као операција своди на операције множења па сабирања, конволуција са тишином ће резултирати тишином. Међутим, конволуциони слојеви имају одступање које се додаје након операције конволуције. Због тога ће излаз из слоја представљати ненулту вриједност, а самим тим ће тај ненулти излаз доћи на улаз следећег слоја. На излазу уметача ће се наћи вјештачки генерисан сигнал од стране свих слојева који се значајно разликује од тишине на улазу. Уметач ће у том случају за задатак имати да ненулти сигнал претвори у тишину што је веома лоше и такво погрешно усвојено знање може утицати на реалне сигнале.

Неколико је разлога због којих је одлучено да сегменти буду ове дужине. Први разлог лежи у чињеници да се модерни процесори боље сналазе са бројевима који су степен двојке чиме се прави мали помак ка оптимизацији. Други разлог мотивисан је дужином поруке која представља водени жиг. Уколико је ова порука прекратка мали су изгледи да ће успјети да се одржи у мрежи уметача која ће евентуално научити да је обрише за вријеме реконструкције. Дужина поруке не би требала бити истог реда величине као дужина сегмента јер ће се тиме угрозити способност уметача да успјешно реконструише сигнал јер порука за мрежу аутоенкодера не представља ништа до адитивни шум. Треће ограничење које је узето у обзир јесте дужина поруке која је одабрана у истраживањима са којим се ово истраживање пореди. Како не постоји ре-

ферентни корпус података било је неопходно обезбиједити сличне услове како би поређење резултата било мјеродавно.

8 Резултати

Не постоји јединствен скуп метода помоћу којих се могу одмјерити перформансе система за уметање воденог жига у аудио сигнале. Наравно, очување квалитета информације и отпорност воденог жига на нападе представљају основ и предуслов за све сложеније методе чије дизајнирање је активно подручје истраживања [79]. У поглављу 8.1 биће објашњена непримјетност, метода за мјерење очувања квалитета, док ће у поглављу 8.2 фокус бити на робустности помоћу које се мјери способност система да водени жиг заштити од напада.

Како би презентовани резултати добили на важности, неопходно их је упоредити са резултатима других алгоритама који су у овој области остварили значајне резултате. Пошто је циљ овог истраживања остварити једнако добре резултате како у непримјетности тако и у робустности, за поређење су одабрани само они алгоритми који остварују задовољавајуће резултате за обје ове мјере.

8.1 Непримјетност

Непримјетност представља способност система за уметање воденог жига да уметне водени жиг тако да се разлика између оригиналног сигнала и сигнала носиоца не може уочити. Не постоји бољи апарат за оцјену квалитета говора од људског слушног система. Међутим, да би те оцјене биле мјеродавне потребно је експертско знање слушалаца. Како се закључци не могу доносити на основу само једне одлуке неопходно је обезбиједити више експерата. Растом групе расте и поузданост резултата. Скала помоћу које се на овај начин врши оцјењивање квалитета говора је MOS скала [80] којом се сигнал оцјењује оцјенама од 1 до 5. Будући да је овај начин оцјењивања захтјеван јер изискује вријеме које ће слушаоци потрошити на слушање и оцјењивање он се обично избјегава како због времена тако и због утрошеног новца. Експертско знање се помоћу аутоматизованих процеса потискује увијек гдје је то могуће, а оцјењивање квалитета говора није изузетак. Тим поводом уводе се аналитичке методе [81] за оцјену квалитета говора које нумеричким упоређивањем оригиналног сигнала са његовим обрађеним паром доносе оцјену о квалитету непримјетности. Како је мишљење људи о резултату ове обраде пресудно, аналитичке методе труде се да опонашају људски слух.

У овом раду користиће се неколико мјера за одређивање непримјетности воденог жига. Прва мјера је однос сигнал-шум (енгл. signal-to-noise ratio -

SNR) и њена употреба је неизбјежна јер представља стандард у дигиталној обради звука. Друга мјера која је одабрана јесте перцептивна евалуација квалитета говора (енгл. perceptual evaluation of speech quality - PESQ) [82]. Ова мјера користи се искључиво код говорних сигнала. Уведена је од стране Интернационалне телекомуникационе уније и представља стандард у тестирању квалитета говора у телефонским системима. За разлику од MOS скале, PESQ вриједности се крећу од -0.5 до 4.5. PESQ даје много бољу процјену квалитета говора у односу на SNR зато што у обзир узима специфичности људског слушног система, али већина истраживања своје резултате и даље презентује употребом SNR мјере.

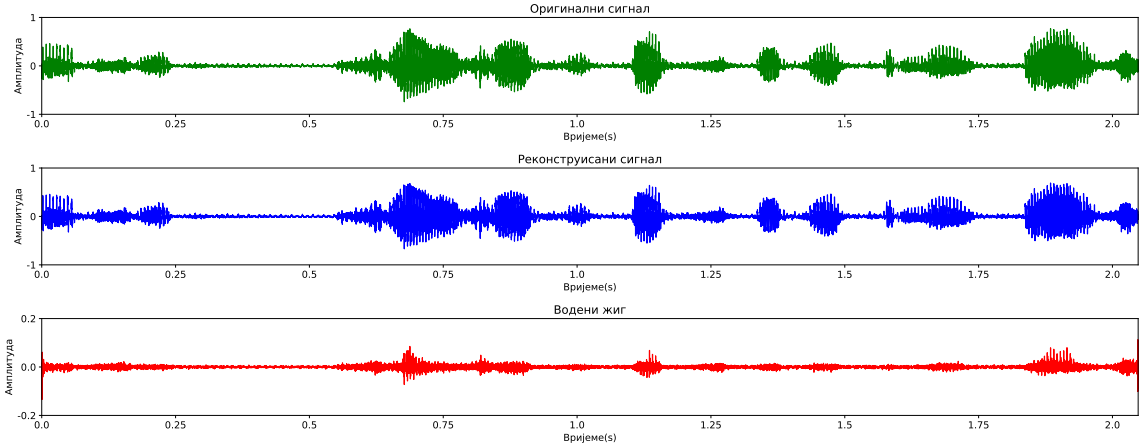
Поређење резултата предложене DNN са најсавременијим техникама у погледу непримјетности дата је у табели 3. Мада је предложени модел побиједио свега двије од пет техника са којима се поредио резултати су охрабрујући јер је добијена вриједност SNR на задовољавајуће високом нивоу. Ова тврдња може се и доказати на примјеру. На слици 22 визуелно се уочава веома мала разлика између оригиналног сигнала (односно једног сегмента) и оног у који је уметнут водени жиг. Одузимањем ова два сигнала добија се њихова разлика која представља грешку тј. шум реконструкције или, симболично, временску репрезентацију воденог жига. Треба имати на уму да је ова разлика јако мала, али да је на графику амплитудом упоредива са сигналом зато што су ради бољег приказа за њено цртање као референтне вриједности амплитуде узете минимална и максимална вриједност разлике, а не сигнала говора. На слици 24 види се увеличан приказ једног малог одсјечка.

Тврдња да разлика оригиналног и реконструисаног сигнала представља временску репрезентацију воденог жига може се оправдати и сликом 23 на којој је приказана амплитуда STFT воденог жига из 22. Уочава се да STFT има јасан облик који не личи шуму и да је водени жиг присутан само у оним интервалима у којима се одвија говор. Такође, водени жиг простире се на свим фреквенцијама сигнала па би га било немогуће обрисати, а да се квалитет значајно не наруши што би се косило са закључцима из поглавља 3.

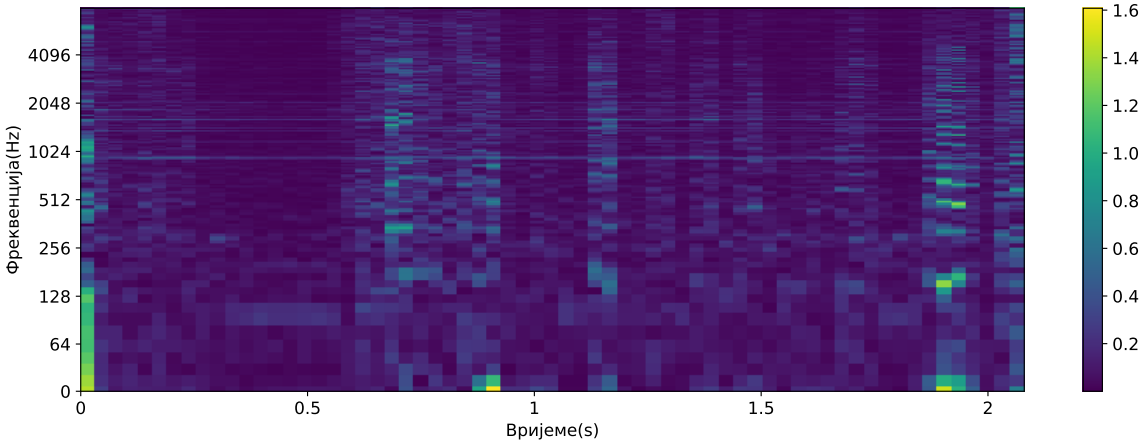
Систем остварује средњу вриједност 4.13 за PESQ што је веома високо и чиме се аналитички доказује способност да успјешно реконструише сигнал, а исти закључак може се донијети и слушањем реконструисаних сигнала.

Метода	SNR (dB)
QDFT-V1 [37]	37.95
[39]	30.18
DCT-b3 [31]	26.44
Предложена DNN	25.41
[17]	22
DWT-IAMM [33]	21.41

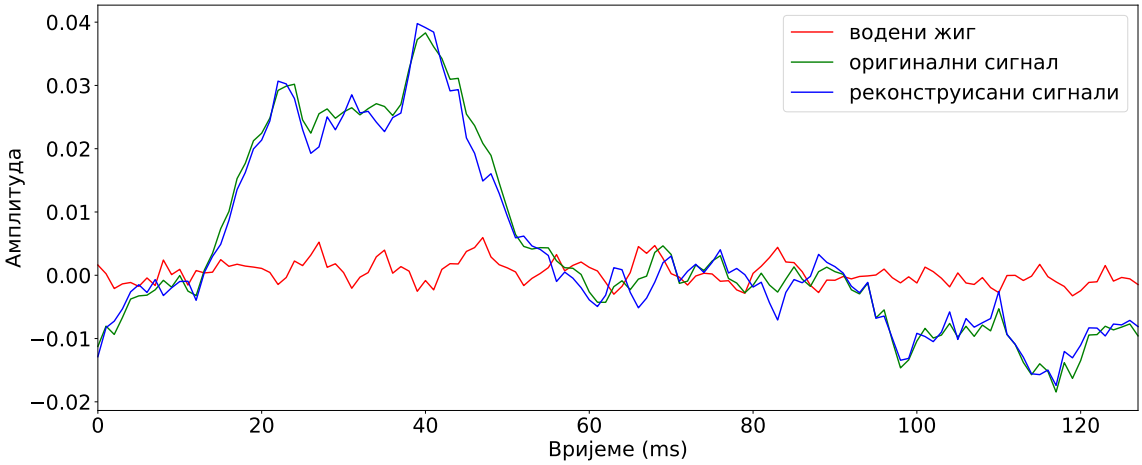
Табела 3: Поређење резултата предложене DNN са најсавременијим техникама у погледу непримјетности



Слика 22: Примјер оригиналног сигнала, његовог реконструисаног пара и њихове разлике која представља водени жиг у временском домену



Слика 23: Амплитуда STFT воденог жига из слике 22



Слика 24: Увеличан приказ једног малог одсјечка из слике 22

Метода	BER (%)				
	Без напада	АГШ	БФ	ПО	АГШ+БФ+ПО
Предложена DNN	0.00	0.01	0.01	0.01	0.05
[17]	0.00	-	0.00	0.00	-
DCT-b3 [31]	0.00	0.09	0.00	-	-
DWT-IAMM [33]	0.00	0.00	0.00	-	-
QDFT-V1 [37]	-	0.07	0.07	-	-
[39]	0.00	0.00	-	0.00	-

Табела 4: Поређење резултата предложене DNN са најсавременијим техникама у погледу робустности. АГШ је адитивни Гаусом шум, БФ је Батервортов филтар, а ПО је пригушивање одбирака

8.2 Робустност

Робустност представља способност воденог жига да пружи отпор нападима чији је задатак да првенствено униште водени жиг, али и да тоураде тако да се квалитет сигнала носиоца не деградира чиме се може компромитовати информација од значаја. Напади представљени у поглављу 3 могу се и комбиновати што се у реалном случају може очекивати мада је мало вјероватно. Како је вјероватноћа да до појединачног напада дође 15%, вјероватноћа да дође до бар једног напада је $\approx 38.6\%$, а вјероватноћа да се сви напади десе истовремено је свега $\approx 0.3\%$. Као мјера робустности кориштен је проценат битова примљених са грешком (енгл. bit error rate - BER) помоћу којег се добија проценат нетачно реконструисаних битова воденог жига. Поређење резултата предложене DNN са најсавременијим техникама у погледу непримјетности дата је у табели 4. Модел остварује јако добре резултате по BER мјери који су упоредиви са конкуренцијом.

9 Закључак

Уметање воденог жига у сигнале говора користи се у одбрани од лажних информација, кршења ауторских права и многих других малициозних радњи. Жртве ових напада могу бити јавне личности као што су музичари или политичари који могу претрпјети непоправљиву штету. Поред јавних личности, на мети ових напада могу се наћи сви појединци, али и читава друштва јер лажне вијести о њима могу створити искривљену слику и нарушити им репутацију. У случају људског гласа, водени жиг за задатак има да прије свега очува кредибилитет информације на начин што ће његово присуство бити јасан доказ да над сигналом нису вршене било какве обраде. Развијањем модерних софтверских алата који се баве дигиталном обрадом сигнала, задатак система за уметање воденог жига бива сложенији.

Предложена DNN која се састоји од уметача и детектора представља систем за уметање воденог жига који настоји да превазиђе све нападе који се могу извршити над сигналом који у себи садржи водени жиг. Уметач и детектор су сами по себи DNN и имају супротстављене задатке и понашају се као супарници такмичећи се једна против друге у остварењу сопствених циљева. Задатак уметача јесте да изврши уметање воденог жига тако да се оствари његова потпуна непримјетност док са друге стране детектор настоји да екстрахује водени жиг из сигнала носиоца па чак и онда када је над сигналом извршен један или серија напада.

За потребе тренирања мреже прикупљен је корпус података сачињен од снимака из Скупштине Црне Горе који је након обраде и изузимања тишине имао преко 800 сати снимљеног говора. За валидацију резултата кориштене су двије метрике, непримјетност и робустност. У случају непримјетности систем остварује SNR вриједности од скоро 26 dB што је упоредиво са најсавременијим приступима у овој области. Додатно, остварене PESQ вриједности као и субјективни закључци након слушања реконструисаних сигнала доводе до закључка да је проблем непримјетности успјешно савладан. Систем је показао да је отпоран на нападе па је тако у случају једног или више напада имао скоро па стопроцентну тачност.

Простора за напредак је доста. Употреба сирових података може се сматрати исправном када су DNN у питању што не значи да ће употреба неког трансформационог домена покварити резултате. Напротив, спектрограм или у овом случају STFT, представља златни стандард у дигиталној обради звука. Међутим, промјена улаза би изискивала значајне промјене на цијелом систему.

Тренирање мреже извршено је на наиван начин. Тежински фактори помоћу којих се омогућила конвергенција резултирају добрим радом мреже, али су одабрани на веома једноставан начин. Посматрањем ових фактора као хиперпараметара мреже олакшало би се проналажење оптималног рјешења.

У случају тренирања мреже у борби против напада вјероватноћа да до напада дође је такође одабрана наивно. Због мале вјероватноће да до једног напада дође, шанса да се сва три напада догоде истовремено је веома мала, а већина серија ни не бива нападнута.

Одабрана вриједност фреквенције одабирања је значајно мања од полазне. Мада је 8 kHz начелно довољно за случај људског говора то неће увијек бити тако. Повећавањем фреквенције повећава се и број одбирака по секунди те се самим тим повећавају и хардверски захтјеви, а и продужава вријеме тренирања. Наравно, овај правац вриједи испитати.

Напади који су имплементирани за потребе овог истраживања су разноврсни и представљају већину приоритетних група напада. Нажалост, ниједан од ових напада заправо не представља десинхронизујуће напада на чему се овом систему може замјерити. Такође, један од напада против којег овај систем није трениран, а који и не представља малициозну радњу јесте пренос сигнала кроз неки медијум и његово поновно снимање. Одличан примјер овог сценарија јесте телефонски разговор или репродукција сигнала на једном уређају те снимање на другом.

Дужина поруке која представља водени жиг утиче директно на рад уметача, али и на рад детектора. Предуга порука и уметач неће извршити добру реконструкцију, прекратка порука никад неће стићи до детектора. Дужина поруке од свега 128 битова резултује малим протоком. Број порука такође утиче на рад обје мреже, а и на коначну оцјену квалитета система. Како је у овом истраживању кориштено свега 8 порука дужине 128 битова постоји простор за напредак.

Литература

[1] Ingemar J. Cox, Joe Kilian, F. Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997.

[2] Fabien A.P. Petitcolas, Ross Anderson, and Markus G. Kuhn. Information hiding-a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999.

[3] Frank Hartung and Martin Kutter. Multimedia watermarking techniques. *Proceedings of the IEEE*, 87(7):1079–1107, 1999.

[4] Nikos Nikolaidis and Ioannis Pitas. Robust image watermarking in the spatial domain. *Signal Processing*, 66(3):385–403, 1998.

[5] Wenjun Zeng and Bede Liu. A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images. *IEEE Transactions on Image Processing*, 8(11):1534–1548, 1999.

[6] Igor Djurović, Srdjan Stanković, and Ioannis Pitas. Digital watermarking in the fractional Fourier transformation domain. *Journal of Network and Computer Applications*, 24(2):167–173, 2001.

[7] Srdjan Stanković, Igor Djurović, and Ioannis Pitas. Watermarking in the space/spatial-frequency domain using two-dimensional Radon-Wigner distribution. *IEEE Transactions on Image Processing*, 10(4):650–658, 2001.

[8] Srdjan Stanković, Igor Djurović, Rainer Herpers, and Ljubiša Stanković. An approach to optimal watermark detection. *AEU - International Journal of Electronics and Communications*, 57(5):355–357, 2003.

[9] Lixin Luo, Zhenyong Chen, Ming Chen, Xiao Zeng, and Zhang Xiong. Reversible image watermarking using interpolation technique. *IEEE Transactions on Information Forensics and Security*, 5(1):187–193, 2010.

[10] Chih-Chin Lai and Cheng-Chih Tsai. Digital image watermarking using discrete wavelet transform and singular value decomposition. *IEEE Transactions on Instrumentation and Measurement*, 59(11):3060–3063, 2010.

[11] Chih-Wei Tang and Hsueh-Ming Hang. A feature-based robust digital image watermarking scheme. *IEEE Transactions on Signal Processing*, 51(4):950–959, 2003.

[12] Frank Hartung and Bernd Girod. Watermarking of uncompressed and compressed video. *Signal Processing*, 66(3):283–301, 1998.

[13] Guang Hua, Jiwu Huang, Yun Q. Shi, Jonathan Goh, and Vrizlynn L.L. Thing. Twenty years of digital audio watermarking—a comprehensive review. *Signal Processing*, 128:222–242, 2016.

[14] Byeong-Seob Ko, Ryouichi Nishimura, and Yoiti Suzuki. Time-spread echo method for digital audio watermarking. *IEEE Transactions on Multimedia*, 7(2):212–221, 2005.

[15] Yong Xiang, Iynkaran Natgunanathan, Dezhong Peng, Wanlei Zhou, and Shui Yu. A dual-channel time-spread echo method for audio watermarking. *IEEE Transactions on Information Forensics and Security*, 7(2):383–392, 2012.

[16] Guang Hua, Jonathan Goh, and Vrizlynn L. L. Thing. Time-spread echo-based audio watermarking with optimized imperceptibility and robustness. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(2):227–239, 2015.

[17] Paraskevi Bassia, Ioannis Pitas, and Nikos Nikolaidis. Robust audio watermarking in the time domain. *IEEE Transactions on Multimedia*, 3(2):232–241, 2001.

[18] Wen-Nung Lie and Li-Chun Chang. Robust and high-quality time-domain audio watermarking based on low-frequency amplitude modification. *IEEE Transactions on Multimedia*, 8(1):46–59, 2006.

[19] Shijun Xiang and Jiwu Huang. Histogram-based audio watermarking against time-scale modification and cropping attacks. *IEEE Transactions on Multimedia*, 9(7):1357–1372, 2007.

[20] Xingyuan Liang and Shijun Xiang. Robust reversible audio watermarking based on high-order difference statistics. *Signal Processing*, 173:107584, 2020.

[21] Robert Devaney. *An introduction to chaotic dynamical systems*. Westview Press, Cambridge, Massachusetts Boulder, Colorado, 2003.

[22] Darko Kirovski and Henrique S. Malvar. Spread-spectrum watermarking of audio signals. *IEEE Transactions on Signal Processing*, 51(4):1020–1033, 2003.

[23] Henrique S. Malvar and Dinei A. Florencio. Improved spread spectrum: a new modulation technique for robust watermarking. *IEEE Transactions on Signal Processing*, 51(4):898–905, 2003.

[24] Wei Li, Xiangyang Xue, and Peizhong Lu. Localized audio watermarking technique robust against time-scale modification. *IEEE Transactions on Multimedia*, 8(1):60–69, 2006.

[25] David Megías, Jordi Serra-Ruiz, and Mehdi Fallahpour. Efficient self-synchronised blind audio watermarking system based on time domain and FFT amplitude modification. *Signal Processing*, 90(12):3078–3092, 2010.

[26] Yong Xiang, Iynkaran Natgunanathan, Dezhong Peng, Guang Hua, and Bo Liu. Spread spectrum audio watermarking using multiple orthogonal PN sequences and variable embedding strengths and polarities. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(3):529–539, 2018.

[27] Brian Chen and Gregory W. Wornell. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47(4):1423–1443, 2001.

[28] Shaoquan Wu, Jiwu Huang, Daren Huang, and Yunqing Shi. Efficiently self-synchronized audio watermarking for assured audio data transmission. *IEEE Transactions on Broadcasting*, 51(1):69–76, 2005.

[29] Xiang-Yang Wang and Hong Zhao. A novel synchronization invariant audio watermarking scheme based on DWT and DCT. *IEEE Transactions on Signal Processing*, 54(12):4835–4840, 2006.

[30] Baiying Lei, Ing Yann Soon, and Ee-Leng Tan. Robust SVD-based audio watermarking scheme with differential evolution optimization. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(11):2368–2378, 2013.

[31] Hwai-Tsu Hu and Ling-Yuan Hsu. Robust, transparent and high-capacity audio watermarking in DCT domain. *Signal Processing*, 109:226–235, 2015.

[32] Min-Jae Hwang, JeeSok Lee, MiSuk Lee, and Hong-Goo Kang. SVD-based adaptive QIM watermarking on stereo audio signals. *IEEE Transactions on Multimedia*, 20(1):45–54, 2018.

[33] Hwai-Tsu Hu and Tung-Tsun Lee. Frame-synchronized blind speech watermarking via improved adaptive mean modulation and perceptual-based additive modulation in DWT domain. *Digital Signal Processing*, 87:75–85, 2019.

[34] Daniel Gruhl, Anthony Lu, and Walter Bender. Echo hiding. In *Information Hiding*, pages 295–315. Springer Berlin Heidelberg, 1996.

[35] Michael Arnold. Audio watermarking: features, applications and algorithms. In *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, volume 2, pages 1013–1016, February 2000.

[36] In-Kwon Yeo and Hyoung Joong Kim. Modified patchwork algorithm: a novel audio watermarking scheme. *IEEE Transactions on Speech and Audio Processing*, 11(4):381–386, 2003.

[37] Euschi Salah, Khaldi Amine, Kafi Redouane, and Kahlessenane Fares. A Fourier transform based audio watermarking algorithm. *Applied Acoustics*, 172:107652, 2021.

[38] Zhenghui Liu, Yuankun Huang, and Jiwu Huang. Patchwork-based audio watermarking robust against de-synchronization and recapturing attacks. *IEEE Transactions on Information Forensics and Security*, 14(5):1171–1180, 2019.

[39] Slami Saadi, Ahmed Merrad, and Ali Benziane. Novel secured scheme for blind audio/speech norm-space watermarking by Arnold algorithm. *Signal Processing*, 154:74–86, 2019.

[40] Xiaojuan Xu, Hong Peng, and Chengyuan He. DWT-based audio watermarking using support vector regression and subsampling. In *Applications of Fuzzy Sets Theory*, pages 136–144. Springer Berlin Heidelberg, 2007.

[41] Lukas Tegendal. Watermarking in audio using deep learning. Master's thesis, Linköping University, 2019.

[42] Seung-Min Mun, Seung-Hun Nam, Haneol Jang, Dongkyu Kim, and Heung-Kyu Lee. Finding robust domain from attacks: A learning framework for blind watermarking. *Neurocomputing*, 337:191–202, 2019.

[43] Haribabu Kandi, Deepak Mishra, and Subrahmanyam R.K. Sai Gorthi. Exploring the learning capabilities of convolutional neural networks for robust image watermarking. *Computers & Security*, 65:247–268, 2017.

[44] Farah Deeba, She Kun, Fayaz Ali Dharejo, Hameer Langah, and Hira Memon. Digital watermarking using deep neural network. *International Journal of Machine Learning and Computing*, 10(2):277–282, 2020.

[45] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *15th European Conference*, pages 682–697, Munich, Germany, September 2018.

[46] Ljubisa Stankovic. *Digital signal processing with selected topics: Adaptive systems, Time-frequency analysis, Sparse signal processing.* Createspace, 2015.

[47] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

[48] Nathanaël Perraudin, Peter Balazs, and Peter L. Søndergaard. A fast griffin-lim algorithm. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–4, 2013.

[49] Martin Steinebach, Fabien A. P. Petitcolas, Frederic Raynal, Jana Dittmann, Caroline Fontaine, Christian Seibel, Nesrine Fates, and Lucilla Ferri. Stirmark benchmark: audio watermarking attacks. In *Proceedings International Conference on Information Technology: Coding and Computing*, pages 49 – 54, May 2001.

[50] Andreas Lang, Jana Dittmann, Ryan Spring, and Claus Vielhauer. Audio watermark attacks: from single to profile attacks. In *Proceedings of the 7th workshop on Multimedia and security - MM&Sec '05*, pages 39–50, January 2005.

[51] Maryam Tanha, Seyed Dawood Sajjadi Torshizi, Mohd Taufik, and Fazirulhisyam Hashim Abdullah. An overview of attacks against digital watermarking and their respective countermeasures. In *Proceedings Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, pages 265–270, June 2012.

[52] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *CoRR*, abs/1706.02515, 2017.

[53] Igor Đurović. *Digitalna obrada slike.* Univerzitet Crne Gore, Elektrotehnički fakultet, Podgorica, 2006.

[54] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[55] Gilbert Strang. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

[56] Steven Roman, S Axler, and FW Gehring. *Advanced linear algebra*, volume 3. Springer, 2005.

[57] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999.

[58] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $\mathrm{O}\left(1/k^2\right)$. In *Doklady Akademii Nauk SSSR*, 1983.

[59] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011.

[60] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[61] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[62] Timothy Dozat. Incorporating nesterov momentum into adam. In *International Conference on Learning Representations 2016*, 2016.

[63] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, July 2015. PMLR.

[64] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[65] Kosta Pavlović, Slavko Kovačević, and Igor Djurović. Speech watermarking using deep neural networks. In *2020 28th Telecommunications Forum (TELFOR)*, pages 1–4, 2020.

[66] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume 9351, pages 234–241, October 2015.

[67] Yann A. LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient BackProp. In *Lecture Notes in Computer Science*, pages 9–48. Springer Berlin Heidelberg, 2012.

[68] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research - Proceedings Track*, 9:249–256, 2010.

[69] Abraham Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, July 1964.

[70] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 26–33, Toulouse, France, July 2001. Association for Computational Linguistics.

[71] Fernando Pereira, Peter Norvig, and Alon Halevy. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(02):8–12, mar 2009.

[72] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[73] Karol J. Piczak. ESC. In *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, October 2015.

[74] John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, Nancy L. Dahlgren, and Victor Zue. TIMIT acoustic-phonetic continuous speech corpus LDC93S1. `https://catalog.ldc.upenn.edu/LDC93S1`, 1993.

[75] The European Broadcasting Union. Sound quality assessment material recordings for subjective tests. `https://tech.ebu.ch/publications/sqamcd`, 2008.

[76] Thomas Baer, Brian Moore, and Karolina Kluk. Effects of low pass filtering on the intelligibility of speech in noise for people with and without dead

regions at high frequencies. *The Journal of the Acoustical Society of America*, 112:1133–44, 2002.

[77] Benjamin Hornsby and Todd Ricketts. The effects of hearing loss on the contribution of high- and low-frequency speech information to speech understanding. *The Journal of the Acoustical Society of America*, 113:1706–17, 2003.

[78] Claude E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, jan 1949.

[79] Patrick Bas and Teddy Furon. A new measure of watermarking security: The effective key length. *IEEE Transactions on Information Forensics and Security*, 8(8):1306–1317, 2013.

[80] IEEE recommended practice for speech quality measurements. *IEEE Transactions on Audio and Electroacoustics*, 17(3):225–246, September 1969.

[81] Yi Hu and Philipos C. Loizou. Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):229–238, 2008.

[82] Antony W. Rix, John G. Beerends, Michael P. Hollier, and Andries P. Hekstra. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 2, pages 749–752 vol.2, February 2001.