SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1798

PRIMJENA STROJNOG UČENJA U SIGURNOSTI SUSTAVA ZA PREPOZNAVANJE GLASA

Domagoj Penić

Zagreb, veljača 2019.

Sažetak

U ovome diplomskom radu prikazan je cjelokupan razvoj sustava za prepoznavanje glasa baziranog na povratnim neuronskim mrežama. Također, razvijene su i metoda koje pokušavaju iskoristiti njihove nedostatke te na taj način ugroziti sigurnost takvih sustava i integritet podataka korištenih u fazi treniranja modela strojnog učenja.

Prvo poglavlje iznosi motivaciju koja je dovela do razvoja sustava za prepoznavanje glasa i metoda koje predstavljaju napade na takav sustav. U drugome poglavlju opisan je proces nastajanja ljudskog govora te na koji način današnja računala shvaćaju govor. Treće poglavlje iznosi pregled metoda koje se koriste u sustavima za prepoznavanje glasa te pregled metoda koje se koriste u sigurnosti takvih sustava, bilo za obranu ili napad. Četvrto poglavlje opisuje skup podataka, odnosno format korištenih podataka te njihova podjela u skupove. Peto poglavlje opisuje izvlačenje značajki zvučnog signala te sustav za prepoznavanje glasa. Šesto poglavlje opisuje suparničke primjere i pokušaje implementacije, zaključno sa sedmim poglavljem gdje je iznesen konačni zaključak.

Sadržaj

Popis oznaka i kratica

engl	engleski
HMM	Skriveni Markovljev model (engl. Hidden Markov Model)
API	aplikacijsko programsko sučelje (engl. Application Programming Interface)
TTS	pretvorba teksta u govor (engl. Text-To-Speech)
DTW	(engl. Dynamic time warping)
SVM	metoda potpornih vektora (engl. Support Vector Machines)
RNN	povratna neuronska mreža (engl. Recurrent Neural Network)
BPTT	propagacija greške unatrag kroz vrijeme (engl. Backpropagation Trough Time)

Popis tablica

Popis slika

1. Uvod

Raspoznavanje govora tehnologija je o kojoj ljudi sanjaju te rade na njoj već desetljećima. Znanstveno-fantastični filmovi samo su od nekih aspekata modernog društva koji su postavljali standarde i očekivanja što je raspoznavanje govora i na koji način bi trebalo koristiti modernome društvu. Bilo je potrebno čekati jako dugo do sofisticiranih sustava za prepoznavanje glasa, sve dok na svoj red nisu došle metode umjetne inteligencije i strojnog učenja, odnosno duboko učenje zajedno s velikim količinama dostupnih podataka. Nakon toga, prepoznavanje glasa, odnosno glasovne naredbe postale su jedan od standardnih načina komunikacije između čovjeka i računala, jer je jednostavno puno učinkovitije s obzirom da čovjek u prosjeku može izreći 150 riječi u minuti, a natipkati samo 40. Glasovni pomoćnici kao što su Siri razvijene od strane kompanije Apple, Amazon-ova Alexa, Google-ov glasovni asistent te razni sustavi ugrađeni u današnje automobile i video igre postali su dio naše svakodnevnice.

Današnji sustavi bazirani na HMM metodi te dubokim povratnim neuronskim mrežama predstavljaju „state-of-art“ rješenja u sustavima za prepoznavanje glasa. Iako takvi sustavi postaju prilično napredni, mogu postati jako ranjivi i podložni napadima ukoliko ih se ne dizajnira jako pažljivo. U ovome radu analizirane su metode generiranja suparničkih primjera koje navode modele strojnog učenja na pogrešnu klasifikaciju, gdje odredišna klasa može biti proizvoljno odabrana od strane napadača. Današnje aplikacije za mobilno bankarstvo podržavaju plaćanje pomoću glasovnih naredbi te bi se za primjer rečenica „Plati Ivanu 20 HRK.“ mogla jako malim, čovjeku nečujnim promjenama, preoblikovati u „Plati Marku 20 HRK“. Suparnički primjeri samo su jedan od aspekata sigurnosti modela strojnog učenja, pogotovo spoznaja da ih je vrlo jednostavno generirati, a tehnike obrane nisu nimalo jednostavne.

Diplomski rad strukturiran je na sljedeći način. U poglavlju 1 dan je uvod u tematiku te motivacija za razvoj sustava za prepoznavanje glasa i metoda suparničkih primjera. Poglavlje 2 opisuje formiranje ljudskog govora te način

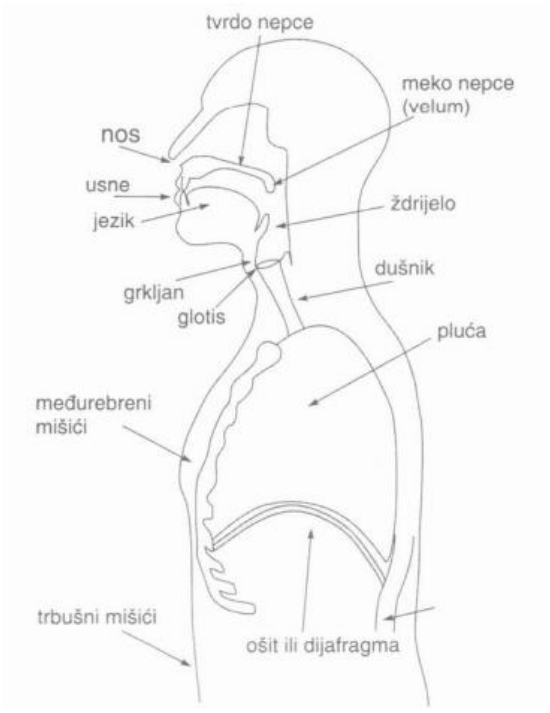
obrade govora od strane računala. Poglavlje 3 iznosi pregled metoda korištenih u sustavima za prepoznavanje glasa, pregled metoda napada na modele strojnog učenja te moguće obrane. Poglavlje 4 opisuje korišteni skup podataka. Poglavlje 5 prezentira razvoj sustava za automatsko prepoznavanje glasa te način izvlačenja značajki iz ljudskog govornog signala. Poglavlje 6 opisuje suparničke primjere te primjenu metoda napada na sustav za prepoznavanje glasa razvijen u poglavlju 5. Na kraju rada iznesen je zaključak te moguće upute za budući rad.

2. Proces nastajanja govora

Ljudi usmeno izražavaju misli, osjećaje i ideje kroz niz složenih pokreta koji mijenjaju i oblikuju osnovni ton stvoren glasom u specifične zvukove koji se mogu dekodirati. Govor se proizvodi precizno koordiniranim djelovanjem mišića u glavi, vratu, prsima i trbuhu. Razvoj govora je postupan proces koji zahtijeva godine prakse.

2.1. Anatomija govornog sustava

Kako bi se napravio model za strojnu obradu govora, potrebno je ponajprije razumjeti kako funkcionira ljudska proizvodnja govora. Općenito, govorni signal je val koji promjenom tlaka zraka putuje od usta govornika do ušiju slušatelja.



Slika 1. Anatomija ljudskog govora

Slika 1 shematski je prikaz anatomije govornog sustava. Pluća proizvode početni tlak zraka koji je bitan za govorni signal. Zatim ždrijelo, usna šupljina i nosna šupljina oblikuju konačni zvuk koji se percipira kao govor.

Ždrijelna i oralna šupljina, zajednički poznate kao vokalni trakt, dinamički se skupljaju te opuštaju, stvarajući sve vrste zvukova kroz rezonancu. Nazalna šupljina otvara još jednu rupu za zrak kako bi stvorila ono što lingvisti nazivaju nazalni glasovi. Zajedno, ove šupljine karakteriziraju zvukove koje ljudi proizvode.

2.2. Formiranje govora

Artikulatorima nazivamo organe koji sudjeluju u formiranju govora koji sadrži informaciju koja se želi prenijeti (jezik, glasnice, itd.). Važnu ulogu u nastajanju govora imaju i pluća govornika koja se pod djelovanjem mišića prsnog koša stišću i potiskuju zrak kroz vokalni trakt.

Značajnu ulogu u procesu formiranja govora imaju glasnice. Smještene su na vrhu dušnika (engl. *trachea*) te djeluju u kombinaciji sa strujanjem zraka. Osciliranje glasnica prilikom strujanja zraka slično je ponašanju piska (engl. *reed*) puhačkih instrumenata. Na frekvenciju titranja glasnica najviše utječu tlak zraka iz pluća te napetost samih glasnica koju je moguće kontrolirati.

Vokalni trakt se ponaša kao svojevrsni filtar koji određuje koje se spektralne komponente pojačavaju a koje prigušuju. Geometrijski je oblik vokalnog trakta naravno promjenjiv, a određuje ga položaj artikulatora govornika.

2.3. Sinteza govora

Gotovo svako moderno računalo ima mogućnost sinteze govora, odnosno pretvorbu pisanog teksta u kompjuterizirani govor. Ovakva tehnologija često dolazi pod akronimom TTS (engl*. Text-To-Speech*) te se može podijeliti u tri faze: pretvorba teksta u riječi, riječi u foneme te fonema u glasove.

Pretvorba teksta u riječi inicijalna je faza sinteze govora te nije nimalo jednostavan zadatak, a obuhvaća smanjenje dvosmislenosti. U ovoj fazi se brojevi, datumi, vremenski trenutci, akronimi te specijalni znakovi pretvaraju u riječi. Također, napada se problem kojeg predstavljaju homonimi, riječi koje se isto pišu, ali se različito izgovaraju te se ispravan naglasak mora odrediti na temelju konteksta u kojemu se riječ nalazi. Na primjer, riječ „duga“ različito se

izgovara u rečenicama „Na nebu se pojavila duga.“ i „Ova rečenica je jako duga.“.

Po definiciji, fonem je najmanja jedinica govora koja je bitna za značenje. Ono što su slova za pisane riječi, to su fonemi govornom jeziku. Zadatak druge faze sinteze govora upravo se bavi pretvorbom svake riječi u njezinu listu fonema. U teoriji, računalo bi moglo imati rječnik gdje bi za svaku moguću riječ imalo listu fonema koji čine njezin izgovor, ali to nije najbolji pristup uzme li se u obzir da taj isti izgovor ovisi o njezinom značenju u tekstu te jedan fonem zvuči različito ovisno koji se fonemi pojavljuju prije i poslije njega unutar iste riječi. Alternativni je pristup rastav riječi na grafeme, najmanje semantički određujuće jedinice pisanog jezika, koji se zatim pretvaraju u foneme na temelju niza jednostavnih pravila.

Završna, treća faza sinteze govora jest pretvorba fonema u zvuk. Postoji više načina koji rješavaju ovaj zadatak, a jedan od najsloženijih je artikulacijska sinteza govora koja se odnosi na modeliranje ljudskog vokalnog trakta, što bi u teoriji trebalo dati glas najsličniji ljudskome govoru.

3. Pregled metoda u sigurnosti sustava za prepoznavanje govora

Model napada na sustav za raspoznavanje govora u strojnom učenju sastoji se od dvije komponente: modela koji je naučen na prikupljenom skupu podataka te od napadača koji različitim tehnikama želi izvući nekakve skrivene značajke modela ili ga navesti na pogrešnu klasifikaciju prethodnom obradom zvučnog signala.

U ovome poglavlju dan je pregled metoda za ostvarenje samoga sustava za prepoznavanje govora te pregled mogućih napada na modele strojnog učenja. Također, pri kraju ovoga poglavlja dan je i kratki pregled mogućih obrana od ovakvih napada.

3.1. Sustavi za prepoznavanje govora

Koncept sustava za raspoznavanje govora kao takvog pojavila se već početkom 11. stoljeća gdje postoje zapisi o navodnom instrumentu koji bi odgovorio s „da“ ili „ne“ na postavljeno pitanje. Bez obzira što ovaj eksperiment nije uključivao nikakvu tehnološku obradu govora, ideja korištenja prirodnog govora kako bi se aktivirala nekakva akcija ostaje kao dio temelja tehnologije za raspoznavanje govora.

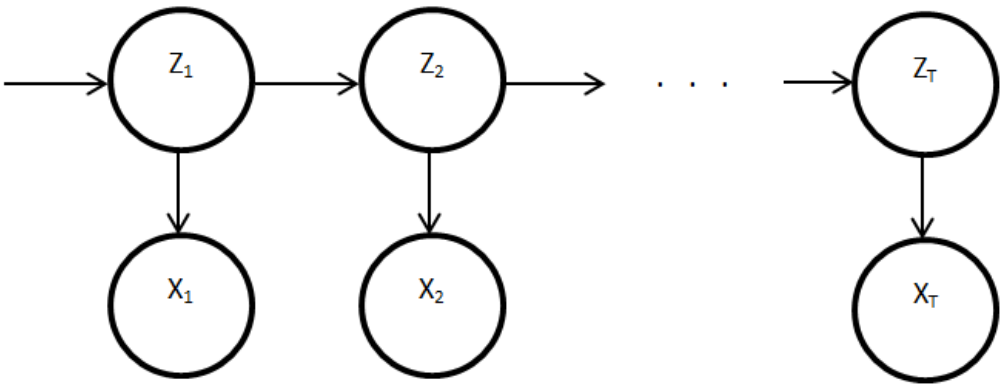
Prvi sustavi za raspoznavanje govora pojavili su se 1950-ih te su mogli raspoznavati jedino znamenke brojevnog sustava. 1960-ih i 1970-ih taj se trend proširio te su se pojavili razni pokušaji i prve kompanije namijenjene upravo komercijalizaciji sustava za raspoznavanje govora.

Slika 2. Pisača mašina za raspoznavanje govora (lijevo) i prva oprema za raspoznavanje kontinuiranog govora (desno)

Do značajnog skoka trebalo je čekati do 1980-ih, kada se, zahvaljujući novim metodama i istraživanjima kako ljudski govor funkcionira, broj riječi koje sustav može prepoznati popeo s nekoliko stotina na nekoliko tisuća. Jedan od glavnih razloga bila je pojava HMM-a.

3.1.1. HMM

HMM statistički je Markovljev model u kojemu se pretpostavlja da je promatrani proces koji se modelira Markovljev proces. HMM zapravo je skup stanja povezanih prijelazima, kako je prikazano u slici 3.



Slika 3. Grafički prikaz HMM-a

Model se sastoji od konačnog skupa skrivenih i promatranih stanja, a svaki čvor $\mathbf{Z_T}$ i $\mathbf{X_T}$ slučajne su varijable. Početak je u određenom početnom stanju te u svakom vremenskom koraku model prelazi u sljedeće stanje, pri čemu je izlazni znak generiran u tom stanju. Odabiri prijelaza i izlaznog znaka su slučajni, određeni razdiobama vjerojatnosti.

Neki od razloga zašto je HMM metoda postala popularna u području raspoznavanja govora jesu statistički i matematički precizni radni okviri, dostupnost algoritama za procjenu parametara modela iz konačnih skupova podataka, fleksibilnost rezultantnog sustava i jednostavnost implementacije cjelokupnog sustava. Iako je HMM dovela performanse sustava za raspoznavanje govora na više razine, ostaju neka temeljna područja u kojima je teorija neadekvatna za govor.

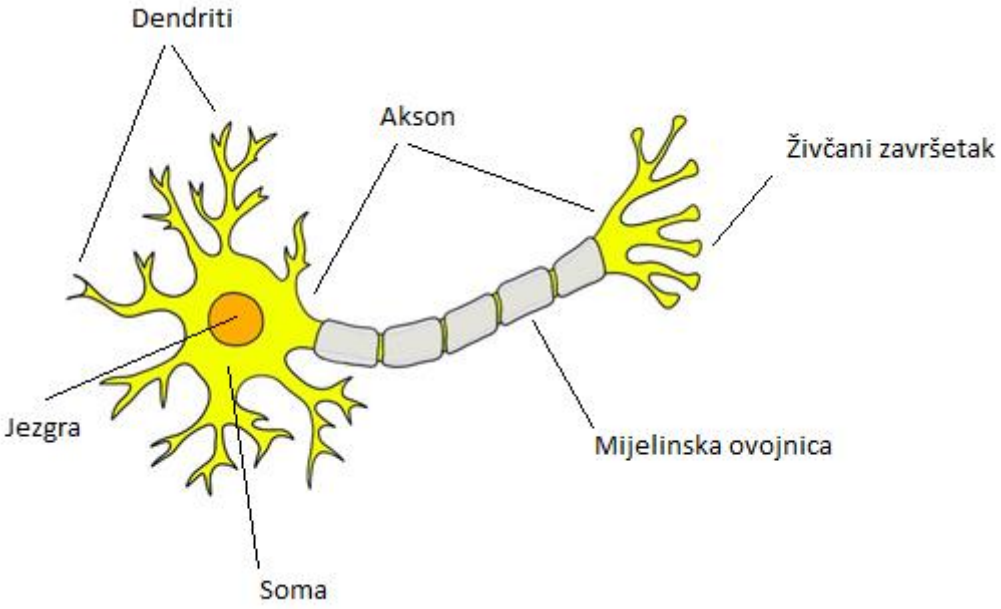
3.1.2. DTW

Jedan od najjednostavnijih načina prepoznavanja jedne riječi jest usporediti je s određenim brojem spremljenih primjeraka te odrediti s kojim primjerkom ima najveću sličnost. Ovaj zadatak može postati jako kompliciran, uzimajući u obzir da iste riječi ne moraju biti istih duljina te da brzina govora može biti različita. DTW (engl. *Dynamic Time Warping*) efikasna je metoda za pronalaženje optimalnog nelinearnog poravnanja dva slijeda.

DTW algoritam uspoređuje parametre nepoznate riječi s parametrima postojećih predložaka svake riječi. Kako bi se poveća točnost ove metode, povećava se broj predložaka, a samim time i vrijeme potrebno za raspoznavanje i memorijski resursi.

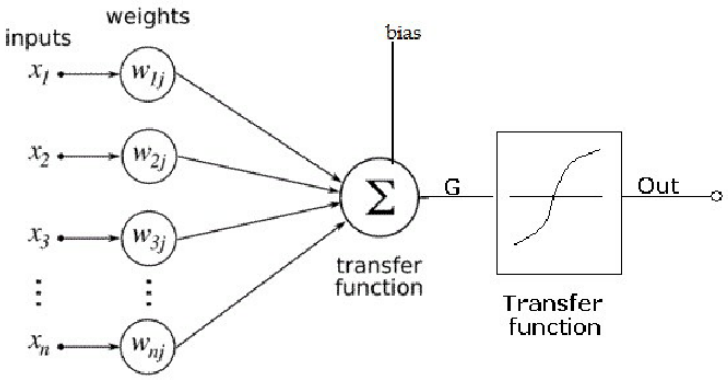
3.1.3. Umjetne neuronske mreže

Neuronska mreža je model zasnovan na načinu rada ljudskog mozga. Ljudski se mozak sastoji od niza gusto povezanih živčanih stanica, neurona. Svaki neuron ima vrlo jednostavnu strukturu, ali skup takvih elemenata predstavlja veliku računalnu moć.



Slika 4. Anatomija neurona

Umjetna neuronska mreža također se sastoji od određenog broja jednostavnih procesnih čvorova, koji su analogni biološkim neuronima u ljudskom mozgu. Neuroni su u mreži povezani težinama kroz koje signal putuje od jednog neurona do drugog. Na slici 5 shematski je prikaz umjetnog neurona: kao ulazni podatak dolazi vektor duljine **n** sa značajkama $\mathbf{x_n}$, koje se množe težinama i zbrajaju zajedno s pristranošću (engl. *bias*) te se ukupan zbroj predaje aktivacijskoj funkciji.



Slika 5. Shematski prikaz umjetnog neurona

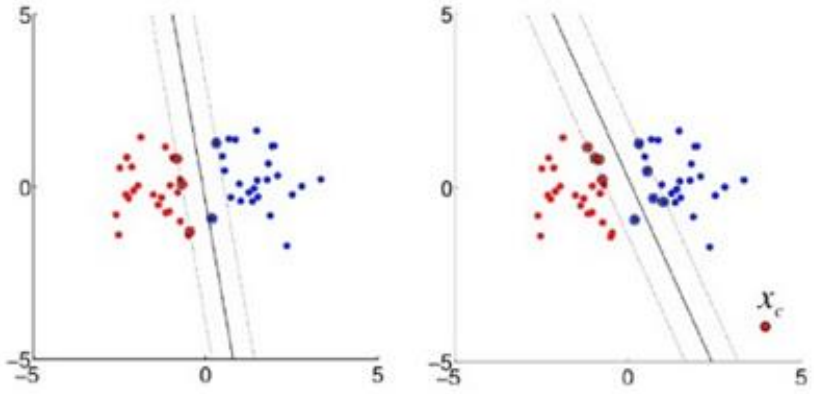
Umjetne neuronske mreže nelinearan su model kojega je, uspoređujući s drugim statističkim modelima, lako za shvatiti i koristiti. Neuronske mreže ne zahtijevaju toliko formalne statističke naobrazbe, nude mogućnost implicitnog detektiranja kompleksnih nelinearnih veza između varijabli te raspon algoritama za treniranje modela. To su samo neki od razloga zašto neuronske mreže predstavljaju učinkovito rješenje u modernih sustavima za raspoznavanje govora te zašto imaju prednost ispred svih drugih algoritama.

3.2. Napadi na modele strojnog učenja

Strojno učenje značajno je napredovalo u zadnjih desetak godina te algoritmi strojnog učenja postižu ljudske rezultate, ili čak i bolje u nekim područjima njihove primjene. Iako spomenuti algoritmi postižu vrhunske rezultate u zadacima kao što su prepoznavanje lica, prepoznavanje znakova te igranje raznih igara, pokazalo se kako su ti isti algoritmi prilično ranjivi. Napadi na modele strojnog učenja mogu se podijeliti u tri skupine: trovanje podataka, tehnike krađe modela te suparnički primjeri.

3.2.1. Trovanje podataka

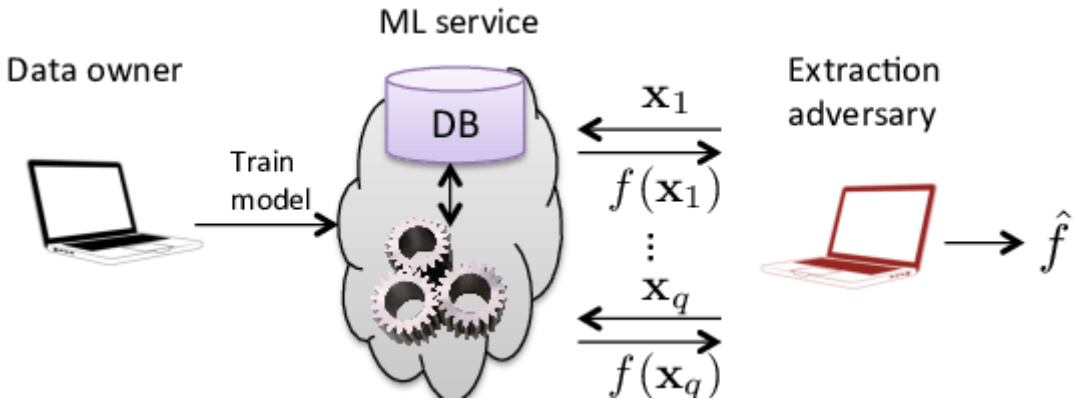
Algoritmi za strojno učenje često su ponovo trenirani na podacima prikupljenim tijekom rada u cilju poboljšanja naučene funkcije distribucije iz koje potječu podaci. Iako se isprva čini razumno rješenje, ponovno treniranje može cijeli sustav dovesti u opasnost. Naime, napadač može modelu predati pažljivo dizajnirane ulazne podatke u cilju pomicanja granice između dviju klasa koja ide njemu u korist. Na slici 6 prikazan je primjer takvog napada: SVM modelu dan je pažljivo izabran ulazni podatak kako bi se decizijska funkcija pomaknula u željenome smjeru.



Slika 6. Iskrivljavanje SVM modela

3.2.2. Tehnika krađe modela

Cilj krađe modela jest „ukrasti“, odnosno duplicirati već naučene modele ili izvući informacije o podacima korištenim za treniranje. Ovakvi napadi privlače veliku pažnju prvenstveno zbog toga što modeli mogu biti trenirani na vrlo osjetljivim podacima neke kompanije (npr. korisničke transakcije, područje kretanja, medicinski podaci)



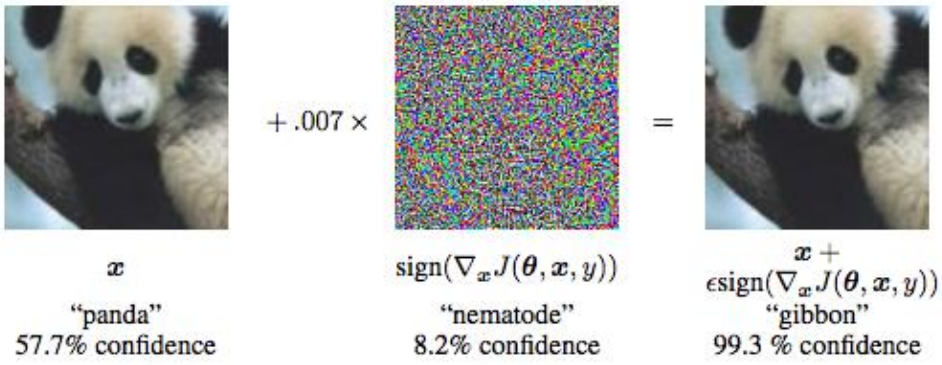
Slika 7. Shema napada krađe modela

Jedan od takvih napada prikazan je na slici 7. Vlasnik podataka te istreniranog modela izlaže API preko kojeg korisnici mogu postavljati proizvoljan broj upita. Napadač postavlja **q** upita te odgovore koristi kako bi proizveo aproksimaciju funkcije **f** funkcijom **f'**.

Druga vrsta napada koji spadaju pod tehnike krađe modela jest izgradnja tzv. „*shadow*“ modela koji napadaču omogućava da za proizvoljni podatak odredi je li on bio korišten u skupu za treniranje ili nije. Iako ovakvi napadi ne pokušavaju rekonstruirati napadnuti model, potencijalno otkrivaju osjetljive informacije.

3.2.3. Suparnički primjeri

Suparnički su primjeri specijalno dizajnirani ulazni podaci s ciljem da ih model pogrešno klasificira, u bilo koju ili proizvoljnu klasu. Neke od primjena ovog napada jest izbjegavanje detekcije email-a kao „*spam*“ ili dizajniranje malicioznih dokumenata koji će izbjeći antivirusne sustave.



Slika 8. Suparnički primjer u klasifikaciji slika

Na slici 8 prikazano je kako se dodavanjem malih smetnji, čovjeku okom (ili uhom, u slučaju raspoznavanja govora) nevidljivih, postiže pogrešna klasifikacija s vrlo velikom pouzdanošću. Počevši od slike pande, dodavanjem malih izračunatih smetnji u izvornu sliku navode model strojnog učenja da je klasificira kao gibona. Ovakvi su se napadi pokazali veoma problematični napadi na duboke modele strojnog učenja. Problem je što ne postoji učinkovit način da model generalizira dobro za sve moguće ulazne podatke. To je vrlo težak zadatak upravo zbog toga što duboke neuronske mreže obavljaju nelinearne optimizacije unutar jako velikih prostora. Suparnički primjeri bit će detaljnije proučeni u kasnijim poglavljima.

3.3. Obrane modela strojnog učenja

Jedan od načina obrane modela strojnog učenja od suparničkih primjera jest treniranje modela i na suparničkim primjerima kako bi model bio što robusniji. Glavna ideja jest generirati suparničke primjere prilikom svakog koraka faze treniranja te ubacivanje istih u polazni skup podataka za treniranje. Pokazalo se kako takvi modeli bolje reagiraju na jednokratne napade, ali da značajno ne pomažu prilikom iterativnih napada.

Razna istraživanja pokušala su razviti binarne klasifikatore bazirane na dubokim neuronskim mrežama kako bi ulazne podatke prepoznale kao suparničke ili „čiste“ prije nego što dođu na ulaz glavnog modela. Ova metoda bi trebala pomoći čak i kada napadač zna da ovakva mreža postoji, ali pokazalo se kako je ovaj pristup i dalje ranjiv na neke vrste napada, kao što je Carlini-Wagner metoda.

Gotovo sve metode obrane pokazala su se učinkovite protiv samo nekih vrsta suparničkih napada te većina adresira probleme u sustavima računalnog vida. Pokazalo se kako ne postoji generalno ispravno rješenje te da će modeli uvijek biti ranjivi na nove i neviđene napade.

4. Skup podataka

Razvoj algoritma za strojno učenje prvenstveno zahtijeva skup podataka. Kako bi model bio što bolji, poželjno je da skup podataka bude raznolik, sakupljen u različitim vremenskim okvirima te od različitih izvora.

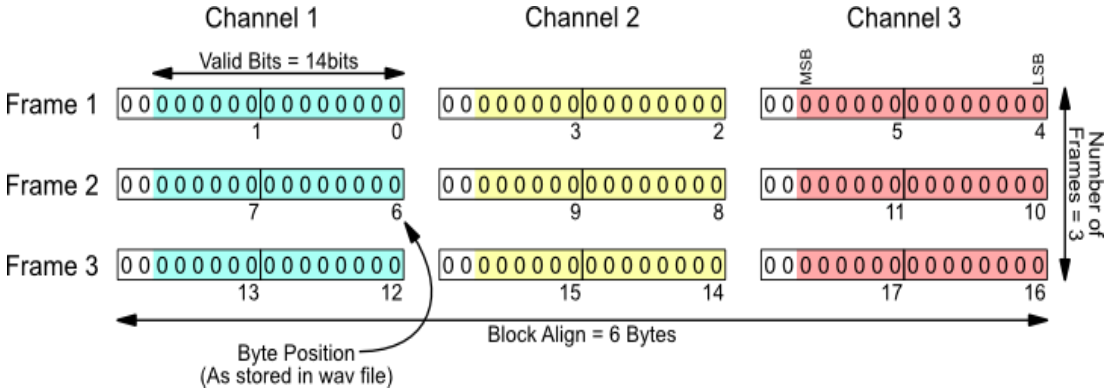
Također, format zapisa bi trebao biti konzistentan kroz cijeli skup podataka te bi to trebalo posebno provjeriti ako se podaci skupljaju od različitih izvora. U ovome radu koriste se zapisi zvučnog signala sakupljeni od različitih govornika engleskog jezika.

4.1. Reprezentacija zvučnog signala

Format zvučnog signala namijenjen je za pohranu zvučnog signala na računalo. Tipovi formata dijele se na:

- Nekomprimirane formate
- Komprimirane formate s gubicima
- Komprimirane formate bez gubitaka

Nekomprimirane datoteke zvučnog zapisa sadrže originalni zapis brojeva sa snimke. Najpoznatiji formati zapisa bez kompresije su WAV i AIFF, a u ovome radu korišten je skup podataka zapisanih WAV formatom.



Slika 9. Struktura WAV formata

WAV poznatiji je kao audio format koji se pretežito koristi na Windows operacijskim sustavima, a izveden je iz RIFF-a (eng. *Resource Interchange*

File Format). Slika 9 ilustrira strukturu odjeljaka za pohranjivanje zvučnih podataka koji se nalaze unutar WAV datoteke. U ovom slučaju postoje 3 audio kanala. Razlučivost je 14 bitova, stoga su za spremanje svakog uzorka potrebna 2 bajta te svaki okvir zahtijeva 6 bajtova, što se naziva *„Block align“*.

4.2. Zvučni zapisi govornih naredbi

U ovome je poglavlju ugrubo opisan skup zvučnih signala korištenih za treniranje modela za raspoznavanje govora, a kasnije i za generiranje suparničkih primjera[1]. Skup podataka sastoji se od malog skupa glasovnih naredbi, izrečenih od različitih govornika te je podijeljen u direktorije sa zvučnim signalima.

Svaki direktorij sadrži isječak glasovne naredbe dužine jedne sekunde, a ime direktorija oznaka je zvučnih isječaka. Svaki direktorij predstavlja jednu brojku engleskog jezika: *„one“*, *„two“*, *„three“*, *„four“*, *„five“*, *„six“*, *„seven“*, *„eight“* i *„nine“*, a svemu što se ne može klasificirati kao jedno od tih kategorija, potrebno je pridijeliti oznaku *„unknown“* ili *„silence“*. Zvučni zapisi zapisani su u slijedećem formatu:

3cfc6b3a_2.wav

Na prvome mjestu nalazi se oznaka govornika, koja je u ovome slučaju 3cfc6b3a. Svaka oznaka je nasumična te bilo kakve informacije o govorniku (spol, godine, lokacija) nisu sačuvane. U svakome direktoriju nalazi se više pojavljivanja glasovne naredbe od istog govornika, te drugi dio formata označava indeks pojavljivanja za tog govornika, gdje u ovome slučaju brojka 2 označava treći zapis istoga govornika.

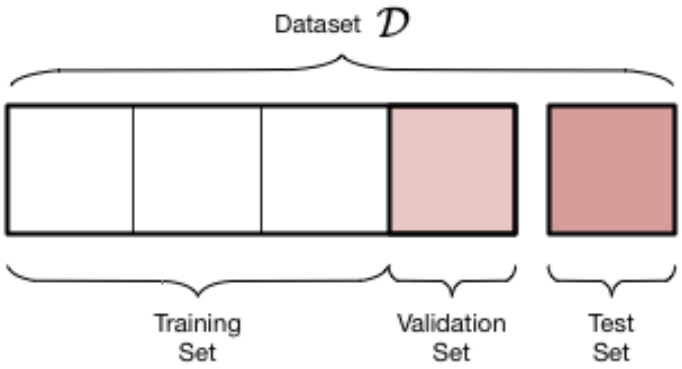
[1] Warden P. Speech Commands: „A public dataset for single-word speech recognition“, 2017. Available from http://download.tensorflow.org/data/speech_commands_v0.01.tar.gz

4.3. Podjela podataka u podskupove

Skup podataka potrebno je, u svrhu poboljšanja generalizacije modela te smanjenja prenaučenosti, podijeliti u različite podskupove:

- skup podataka za treniranje,
- validacijski skup,
- skup za testiranje

Skup podataka za treniranje jest skup koje model vidi tijekom faze treniranja te na kojemu uči. S druge strane, skup za validaciju jest skup koji služi za povremenu evaluaciju modela. Model vidi te podatke, ali ne uči od njih, stoga taj skup služi za postavljanje hiperparametara. Na kraju, model se evaluira točno jednom na skupu za testiranje jednom kada je potpuno istreniran. Skup za testiranje sadrži pažljivo odabrane podatke koji obuhvaćaju veliki raspon različitih klasa s kojima bi se model mogao susresti u stvarnome svijetu.



Slika 10. Raspodjela skupa podataka

Zvučni signali iz skupa podataka kao takvi nisu podijeljeni u skupove podataka za treniranje, validacijski skup te testni skup. Način na koji podijeliti skup podataka ovisi ponajprije o ukupnom broju primjeraka u skupu podataka te o modelu koji se trenira. Na primjer, ukoliko model ima puno hiperparametara, bit će potreban veći validacijski skup.

U svrhu podjele podataka u spomenute skupove, izrađena je komandno-linijska aplikacija u programskom jeziku Java. Za svaki glasovni zapis iz skupa

podataka određuje se njegova pripadnost skupu, a ona je opisana enumeracijom DatasetType:

```
public enum DatasetType {

    TRAINING,
    VALIDATION,
    TEST

}
```

Slika 11. Enumeracija „DatasetType“

Klasa DatasetPartitioner enkapsulira logiku koja na temelju imena datoteke glasovnog zapisa određuje kojem skupu ona pripada. Kako bi prilikom svakog izvođenja programa, odnosno prilikom svake nove podjele podataka (npr. dodavanje novih zapisa) svaki zapis ostao u onom skupu podataka u koji je prethodni put bio dodijeljen, korištena je ugradbena funkcija sažetka *(*engl. *hash*) koja na temelju imena datoteke vraća cijeli broj. Također, želimo da govorni zapisi istih govornika završe u istom skupu, stoga prilikom primjene funkcije sažetka u obzir nije uzeto zadnjih 6 znakova imena datoteke („_“, indeks zapisa, te ekstenzija „.*wav*“).

```
class DatasetPartitioner {

    private static final int MAX_PER_CLASS = (2 << 26) - 1;

    private double validationPercentage;
    private double testPercentage;

    DatasetPartitioner(double validationPercenetage, double testPercentage) {
        this.validationPercentage = validationPercenetage;
        this.testPercentage = testPercentage;
    }

    DatasetType getDataset(File datasetFile) {
        String fileName = datasetFile.getAbsolutePath();
        // Remove file extension and record index
        String fileNameNoSuffix = fileName.substring(0, fileName.length() - 6);

        double percentage = Math.abs((fileNameNoSuffix.hashCode() % (MAX_PER_CLASS + 1))
                * (100.0 / MAX_PER_CLASS));

        if (percentage < validationPercentage) {
            return DatasetType.VALIDATION;
        } else if (percentage < (validationPercentage + testPercentage)) {
            return DatasetType.TEST;
        } else {
            return DatasetType.TRAINING;
        }
    }

}
```

Slika 12. Klasa „DatasetPartitioner“

Skup podataka od 23673 glasovna zapisa podijeljen na način prikazan u tablici 1.

Skup podataka	Postotak	Broj zapisa
Treniranje	0.8	19036
Validacija	0.1	2376
Testiranje	0.1	2261

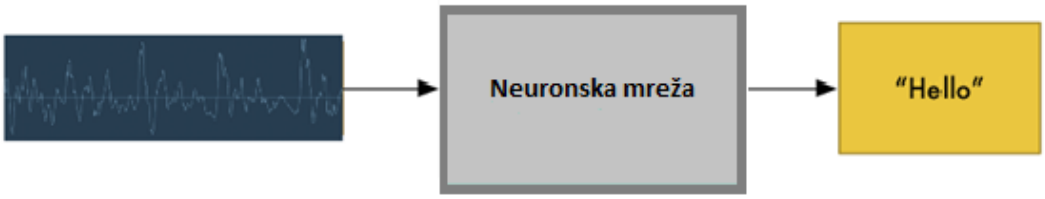
Tablica 1. Raspodjela skupa podataka

Skup podataka podijeljen je na sljedeći način: 80% podataka korišteno je u fazi treniranja, a otprilike 10% podataka korišteno je za validaciju, odnosno za završno testiranje.

5. Automatsko raspoznavanje govora

Automatsko raspoznavanje govora je skup računalnog hardvera i softverskih algoritama namijenjenih za identifikaciju i obradu ljudskog glasa. Koristi se za identifikaciju riječi izrečenih od strane govornika ili za provjeru autentičnosti identiteta osobe koja govori u sustav (biometrijska provjera autentičnosti).

Raspoznavanje govora prvenstveno se koristi za pretvaranje izgovorenih riječi u tekst spremljen na računalu. U pravilu, sustav zahtijeva unaprijed konfigurirane ili spremljene glasove primarnih korisnika. Ljudi trebaju trenirati takav sustav pohranjivanjem obrazaca govora i rječnika u sustav.



Slika 13. Sustav za raspoznavanje govora

Na slici 13 ilustriran je primjer sustava za raspoznavanje govora koji je baziran na neuronskoj mreži. Čovjek proizvodi zvučni signal te ga računalo registrira i sprema u svoju radnu memoriju. Iz zvučnog signala izvlače se značajke koje najbolje opisuju taj signal te se takav predaje prethodno naučenom modelu neuronske mreže koja kao izlaz daje niz tekstualnih znakova.

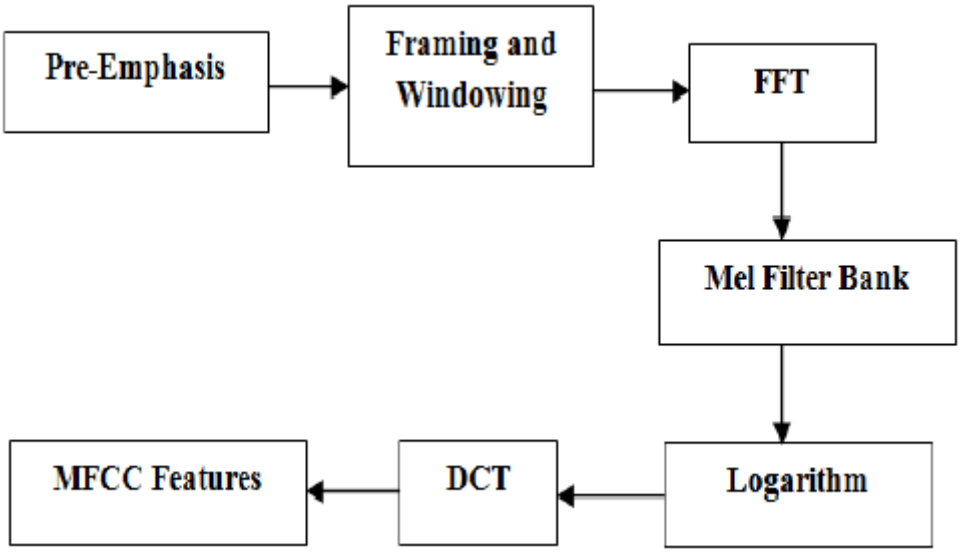
5.1. MFCC analiza

Prvi korak u svakom sustavu za automatsko raspoznavanje govora jest izvući značajke, tj. identificirati komponente audio signala koje su dobre za identificiranje jezičnog sadržaja i odbacivanje svih drugih stvari koje nose informacije kao što su pozadinska buka, emocije itd.

Jedna od najvažnijih stvari koje treba razumjeti u govoru jest da se zvukovi koje generira čovjek filtriraju po obliku vokalnog trakta, uključujući jezik, zube itd. Ovaj oblik određuje zvuk koji nastaje te ukoliko možemo točno odrediti oblik, to bi trebalo dati točnu reprezentaciju fonema koji se proizvode. Oblik vokalnog trakta očituje se u ovojnici kratkotrajnog spektra snage, a zadatak MFCC analize jest što točnije predstaviti ovu ovojnicu.

5.1.1. Provođenje MFCC analize

Blok dijagram provođenja MFCC analize prikazan je slikom 14.



Slika 14. Koraci provođenja MFCC analize

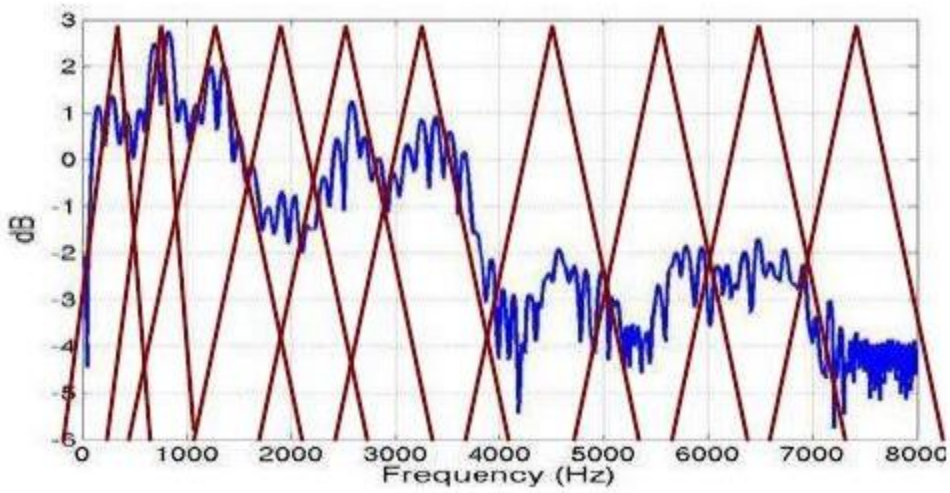
Prvi blok u dijagramu (engl. *pre-emphasis*) služi za pojačavanje komponenti visokih frekvencija tj. pojačavanje energije dijelova signala koji su viših frekvencija. Uzorci se zvuka zatim formiraju u okvire uobičajenog trajanja 20-25 milisekundi. Okviri se množe vremenskim otvorom za kojeg se obično koristi Hammingov vremenski otvor.

Proizvodnja govora opisuje se preko modela izvor-filtar. Izvor se povezuje sa zrakom koji se ispušta kroz pluća, a filtar daje oblik spektru signala provodeći tako različite zvukove. Vokalni trakt opisuje se spomenutim filtrom te dobre reprezentacije govornog signala pokušavaju ukloniti utjecaj izvora, a

ostvariti dobru karakterizaciju filtra, kako bi sustav dao isti odziv npr. za visoki „*pitch“* ženskog glasa i niski „*pitch“* muškog glasa.

5.1.2. Mel-Spektar

Mel-Frequency analiza govora zasnovana je na eksperimentima ljudskog uha gdje je uočeno da se ljudsko uho ponaša kao neka vrsta filtra tj. koncentrira se samo na određene frekvencijske komponente. Takvi su filtri neuniformno raspoređeni po frekvencijskoj osi. Na slici 15 prikazan je raspored takvih filtara.



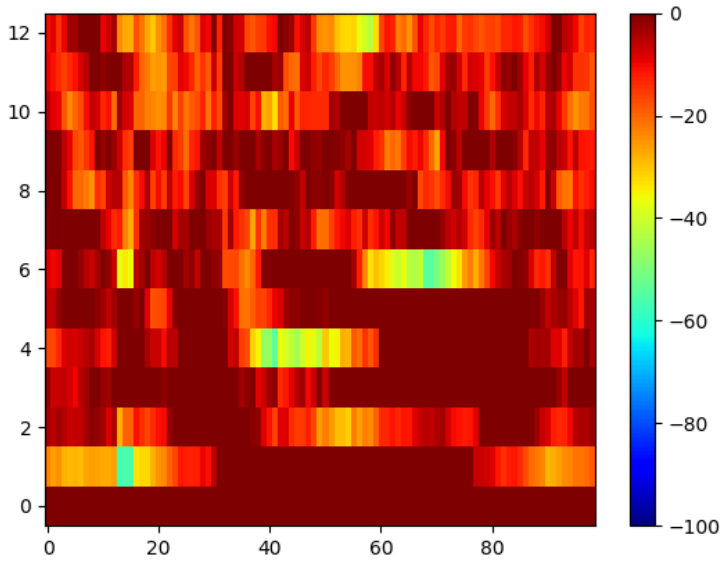
Slika 15. Primjer filtara u MFCC analizi

Spektar koji je prošao kroz ovakve filtre naziva se Mel-Spektar. Ukoliko se nad ovakvim spektrom provede Kepstralna analiza, dobiveni kepstralni koeficijenti nazivaju se Mel-Frequency kepstralni koeficijenti. MFCC analiza predstavlja „*state-of-art*“ u sustavima za automatsko raspoznavanje govora.

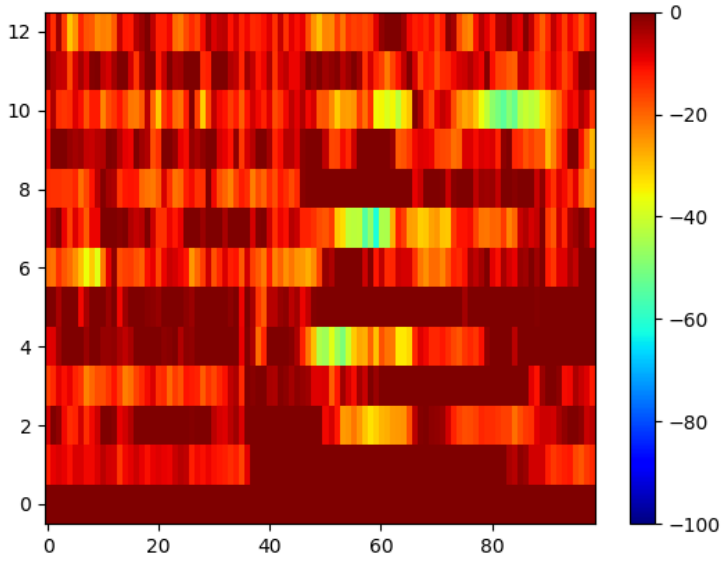
5.1.3. Primjer značajki dobivenih MFCC analizom

Kao što je već spomenuto, u ovome radu korišten je skup zvučnih signala duljine jedne sekunde. Uzme li se u obzir da je duljina vremenskog okvira MFCC analize 20ms s pomakom od 10ms, kao rezultat analize dobije se matrica dimenzija 13 x 99. Za svaki promatrani vremenski okvir, dobije se 13 koeficijenata.

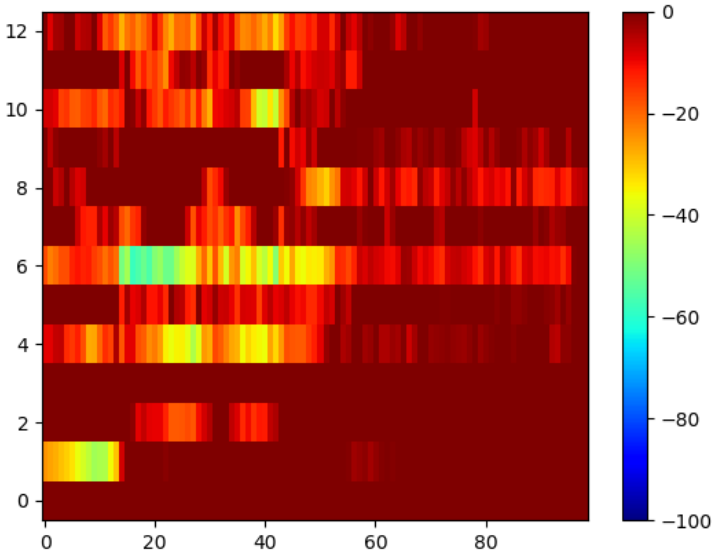
Na slijedećim slikama kao primjer prikazane su vrijednosti MFCC koeficijenata za muške i ženske glasove koji izgovaraju engleske riječi *„one“* i *„seven“*. Na osi x nalaze se redni brojevi vremenskih okvira, što ujedno predstavlja i vrijeme, a na osi y redni brojevi MFCC koeficijenata.



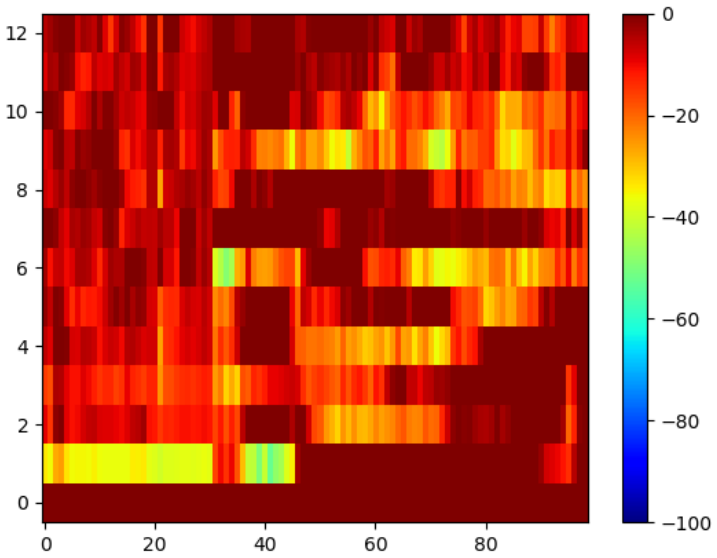
Slika 16. Muški govornik, izgovor riječi "*one*"



Slika 17. Ženski govornik, izgovor riječi "*one*"



Slika 18. Muški govornik, izgovor riječi "*seven*"



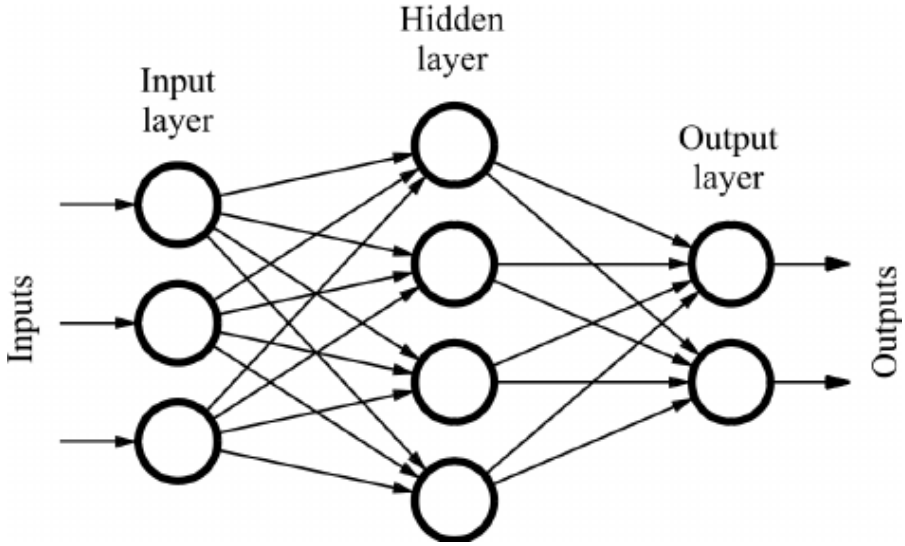
Slika 19. Ženski govornik, izgovor riječi "*seven*"

5.2. Neuronske mreže u sustavu za raspoznavanje govora

Povratne neuronske mreže i njihova varijacija LSTM pokazali su se kao snažno oružje u klasifikaciji sekvencijalnih podataka te su stoga i korišteni u ovome radu kao glavni dio sustava za raspoznavanje govora.

5.2.1. Unaprijedne neuronske mreže

Povratne neuronske mreže nadogradnja su na postojeće unaprijedne neuronske mreže. Obje vrste neuronskih mreža nazvane su po načinu na koje informacije teku nizom matematičkih operacija koje se odvijaju u čvorovima mreža. Na slici 20 prikazan je primjer arhitekture unaprijedne neuronske mreže s jednim skrivenim slojem, gdje je tok podataka s lijeva na desno kroz umjetne neurone povezane težinama, koji su bili opisani u prethodnim poglavljima.



Slika 20. Arhitektura neuronske mreže s jednim skrivenim slojem

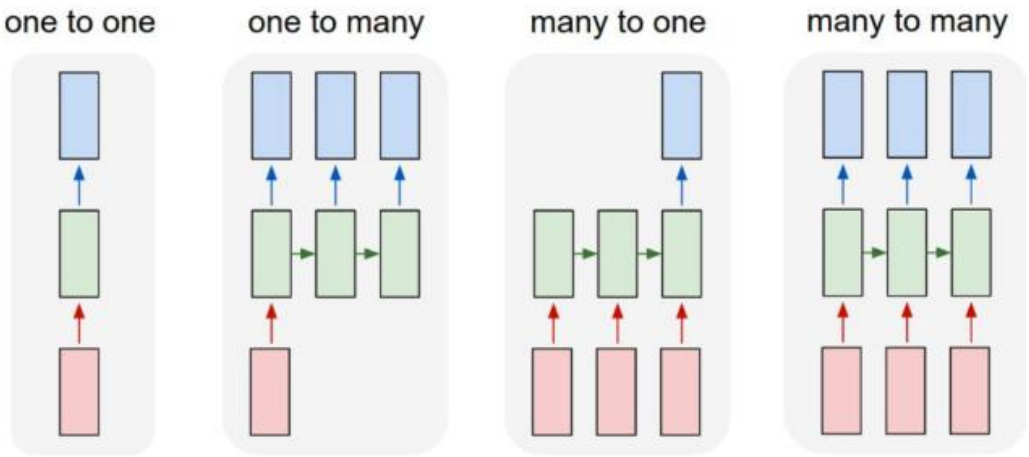
U slučaju unaprijednih mreža, ulazni podaci transformirani su u izlazne oznake, odnosno klasu u koju ulazni podatak pripada. Treniranje takvih mreža odvija se nadziranim učenjem s označenim podacima sve dok se ne minimizira pogreška prilikom pogađanja njihovih klasa. S treniranim skupom parametara (ili težinama koje skupno predstavljaju model), mreža pokušava klasificirati podatke koje nikada nije vidjela.

Nedostatak unaprijednih neuronskih mreža je to što nemaju pojam o vremenu, odnosno redoslijedu, a jedini ulazni podatak koji se promatra jest trenutni primjer kojem je mreža izložena. Takve mreže zaboravljaju svoju nedavnu prošlost, a nostalgično se prisjećaju samo trenutaka iz procesa treniranja.

5.2.2. Povratne neuronske mreže

Povratne neuronske mreže relativno su stari izum, kao i većina drugih algoritama dubokog učenja. Pojavile su se 1980-ih, ali su svoj potencijal počele pokazivati tek prije nekoliko godina, upravo zbog povećanja računalne moći i količini dostupnih podataka koje danas posjedujemo.

Zbog svoje interne memorije, povratne neuronske mreže imaju mogućnost zapamtiti važne informacije o ulaznim podacima, što im omogućava da vrlo precizno predvide što će se sljedeće dogoditi.

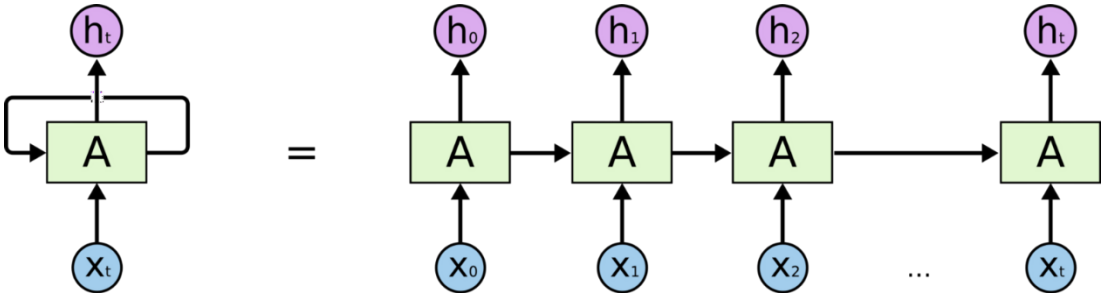


Slika 21. Varijacije dubokih povratnih neuronskih mreža

Unaprijedne neuronske mreže na temelju jednog ulaznog podataka generiraju jedan izlaz, dok povratne neuronske mreže imaju dodatne mogućnosti preslikavanja „*one-to-many*“, „*many-to-one*“ i „*many-to-many*“, kako je to prikazano na slici 21. Na primjer, preslikavanje „*many-to-one*“ koristi se u slučaju kada se na temelju ulaznog teksta pokušava predvidjeti

emocija koja prevladava u tekstu. Ulazni tekst varijabilnog je broja riječi, a rezultat klasifikacije je samo jedna oznaka.

Dok informacija kod unaprijednih neuronskih mreža teče ravno kroz sve čvorove, kod povratnih neuronskih mreža svaki se čvor posjećuje više puta upravo zbog povratnih veza koje čine petlju. S druge strane, povratne neuronske mreže ne dobivaju kao ulaz samo trenutni ulazni podatak, nego je njihovo perceptivno polje prošireno ulaznim podacima iz prošlosti, odnosno već obrađenim elementima.



Slika 22. „Odmotavanje“ povratne neuronske mreže

Odluka ovakve neuronske mreže u vremenskom trenutku *t* ovisi o odluci te iste neuronske mreže donesene u trenutku *t-1*. Stoga povratne neuronske mreže imaju dva izvora ulaznih podataka, sadašnjost i nedavna prošlost, koje kombiniraju kako bi donijele odluku za nove podatke. Prilično slično kako to ljudi rade i u stvarnom životu.

Kao što je već prije spomenuto, povratne neuronske mreže razlikuju se od unaprijednih po povratnoj petlji koja povezuje mrežu s njezinim prethodnim odlukama, unoseći vlastite izlaze kako ulaze nakon svakog trenutka. Stoga se često kaže da povratne neuronske mreže imaju memoriju, a dodavanje memorije neuronskoj mreži ima svrhu. U samoj sekvenci ulaznih podataka postoje određene informacije te ih povratne neuronske mreže iskorištavaju u donošenju svojih odluka, dok unaprijedne mreže to ne mogu.

Spomenuta informacija redoslijeda ulaznih podataka sačuvana je u skrivenom stanju povratne neuronske mreže. Matematički se proces prenošenja memorije unaprijed može prezentirati sljedećom formulom:

$$h_t = \phi(Wx_t + Uh_{t-1})$$

Skriveno stanje u vremenskom trenutku ***t*** označeno je oznakom $\boldsymbol{h_t}$. Kao što se može vidjeti iz formule, skriveno stanje ovisi o ulaznom podatku u trenutnom vremenskom koraku $\boldsymbol{x_t}$, pomnoženom s matricom težina ***W***, čemu se pribraja skriveno stanje prijašnjeg vremenskog trenutka $\boldsymbol{h_{t-1}}$ pomnoženo matricom ***U***. Matrice težina ***W*** i ***U*** predstavljaju filtere, odnosno koliku ćemo važnost pridijeliti ulaznom podatku u trenutnom vremenskom koraku, a koliku skrivenom stanju iz prethodnog koraka. Na dobivenu sumu primjenjuje se funkcija ***Φ***, koja je najčešće sigmoida ili tangens hiperbolni.

5.2.3. Propagacija unatrag kroz vrijeme

Svrha povratnih neuronskih mreža jest precizno klasificiranje sekvencijalnih ulaza te u svrhu toga treniranje takvih mreža se oslanja na učenje širenjem pogreške unatrag (engl. *backpropagation*) i gradijentni spust.

Povratne neuronske mreže oslanjaju se na proširenu verziju širenja pogreške unatrag zvanu širenje pogreške unazad kroz vrijeme, što je ništa više nego naziv za primjenu algoritma širenja pogreške unazad na odmotanu povratnu neuronsku mrežu.

```
// a[t] - ulaz u trenutku t
// y[t] - izlaz u trenutku
back_propagation_trough_time(a, y)
    odmotaj mrezu da sadrzi k instanci od f
    dok nije ispunjen uvjet zaustavljanja:
        x = nul-vektor
        za t od 0 do n - k:
            ulaze mreze postavi na x, a[t], a[t + 1], ..., a[t + k - 1]
            p = unaprijedni prolaz kroz mrezu
            error = y[t + k] - p
            propagiraj gresku unatrag
            azuriraj tezine
            x = f(x, a[t])
```

Slika 23. Pseudokod BPTT algoritma

Kada se povratna neuronska mreža „odmota“, može se promatrati kao vrlo široka unaprijedna neuronska mreža, kako je to prikazano na slici 22. S takvom arhitekturom pojavljuju se i problemi nestajućih, odnosno eksplodirajućih gradijenata. Također, za vrlo dugačke sekvence podataka propagacija unatrag kroz vrijeme postaje vrlo računalno zahtjevna, stoga se u praksi koristi varijanta BPTT-a zvana skraćeno širenje pogreške unatrag kroz vrijeme (engl. *Truncated backpropagation trough time*). Skraćeno širenje pogreške unatrag aproksimacija je potpunog širenja namijenjena za jako dugačke sekvence, a glavna ideja jest svakih k1 koraka primijeniti BPTT k2 koraka unatrag.

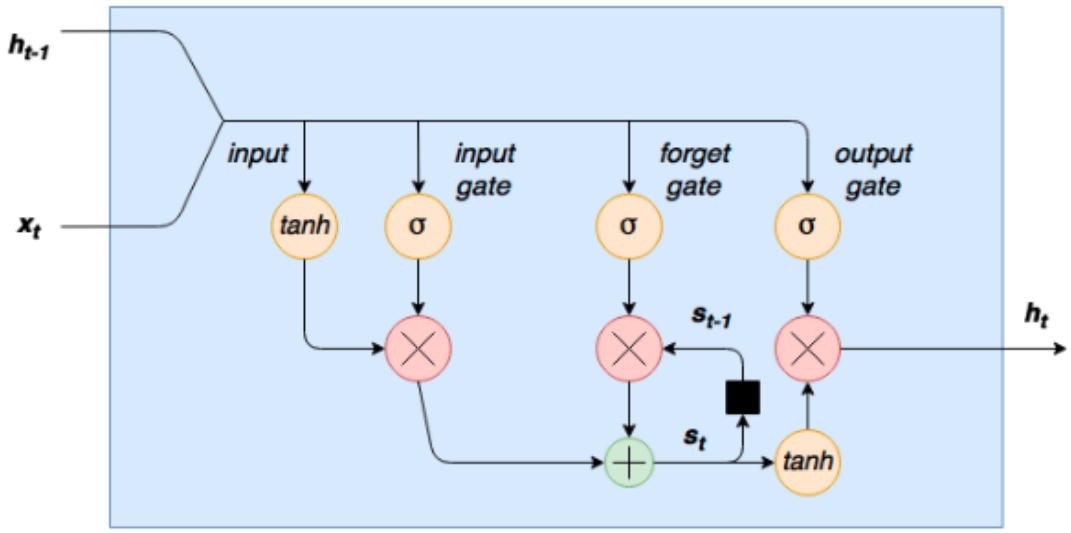
5.3. LSTM

Jedan od glavnih razloga zašto su povratne neuronske mreže privlačne jest ideja povezivanje prethodnih informacija s trenutnim ulaznim podacima. Na primjer, u sustavu za automatsko raspoznavanje govora prilikom prepoznavanju trenutne riječi informacija o prethodno prepoznatim riječima može puno pridonijeti. Prvotna arhitektura povratnih neuronskih mreža u teoriji podržava učenje gdje su relevantne informacije relativno udaljene od trenutnog ulaznog podatka, ali u praksi se pokazalo kako takve mreže imaju kratkotrajno pamćenje.

Sredinom devedesetih pojavila se varijanta povratnih neuronskih mreža pod imenom LSTM (engl. *Long Short-Term Memory units*). LSTM neuronske mreže proširenje su povratnih neuronskih mreža, gdje se zapravo povećava njihova memorija te se rješava problem učenja dugoročnih ovisnosti.

5.3.1. Struktura LSTM ćelije

Struktura LSTM ćelije prikazana je slikom 23.



Slika 24. Unutarnja struktura LSTM ćelije

Protok podataka je s lijeva na desno, gdje je trenutni ulazni podatak označen s $\mathbf{x_t}$, a izlaz prethodne ćelije s $\mathbf{h_{t-1}}$ koji su spojeni u jedan ulazni podatak te se kao takvi šalju na ulaze u različita LSTM vrata.

5.3.1.1. „Input gate“

Prvo, ulaz je preslikan na interval između -1 i 1 koristeći aktivacijsku funkciju tangens hiperbolni. To se može prikazati slijedećom formulom:

$$g = \tanh(b^g + x_t U^g + h_{t-1} V^g)$$

Gdje su **Ug** i **Vg** težine za ulaz i prethodni izlaz ćelije.

Preslikani je ulaz zatim pomnožen s izlazom „*input gate*“-a. „*Input gate*“ zapravo je skriveni sloj čvorova aktiviranih sigmoidom, koji kao izlaz daje vrijednosti u intervalu između 0 i 1 te zatim pomnožen s ulazom zapravo određuje koji su ulazni elementi „upaljeni“, a koji „ugašeni“. Formula kojim se to može sumirati je:

$$i = \sigma(b^i + x_t U^i + h_{t-1} V^i)$$

Izlaz ulazne faze LSTM ćelije može se prikazati slijedećom formulom, gdje operator ∘ predstavlja množenje po elementima:

$$g \circ i$$

5.3.1.2. „Forget gate“ i skriveno stanje

Dugoročno pamćene LSTM mreže sadržano je u unutar ćelija u tzv. skrivenim stanjima. Skriveno stanje LSTM ćelije modificirano je izlazom „forget gate“-a, koji se može prikazati sljedećom jednadžbom:

$$f = \sigma(b^i + x_t U^i + h_{t-1} V^i)$$

Ukoliko je vrijednost izlaza **f** jednaka jedan, to znači da će se sve informacije zadržati u skrivenom stanju u tom koraku, a što je ta vrijednost bliža 0, to se informacije više „zaboravljaju“.

5.3.1.3. „Output gate“

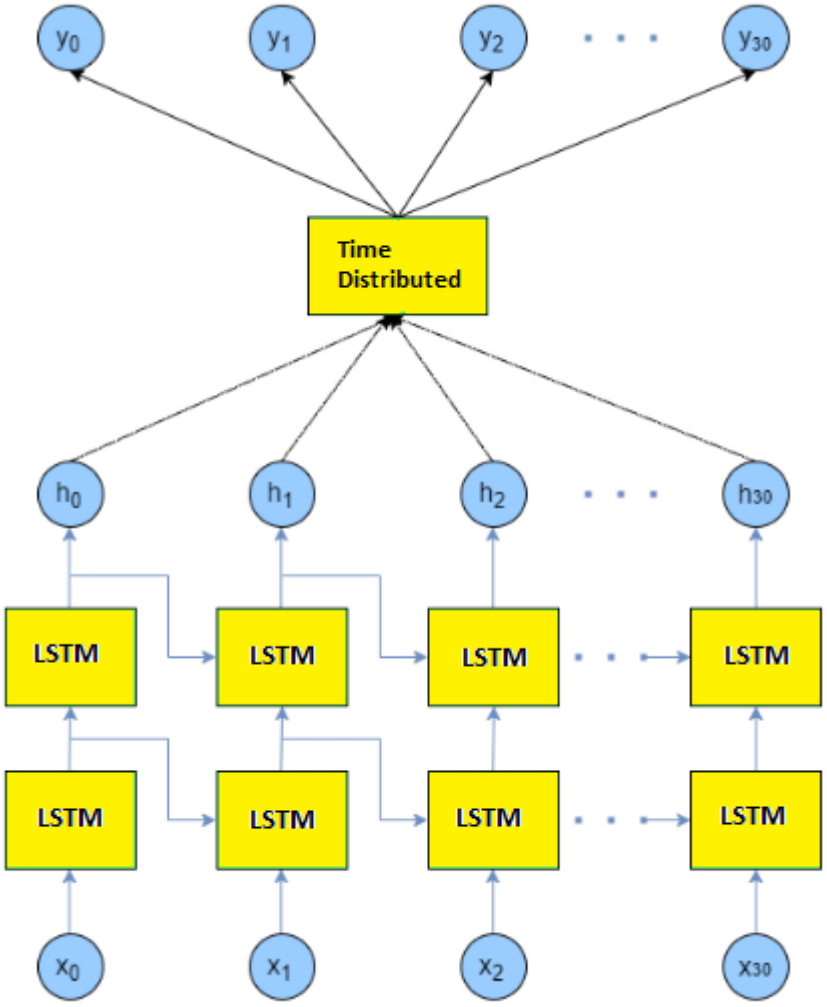
„Output gate“ određuje što će biti sljedeće skriveno stanje, a ono sadrži informacije o prethodnim ulaznim podacima. Prvo, prethodno skriveno stanje zajedno s trenutnim ulazom se predaje sigmoidalnoj funkciji, a zatim tangensu hiperbolnom kako bi se dobila vrijednost na intervalu između -1 i 1 te se takvo prenosi sljedećem vremenskom trenutku.

$$o = \sigma(b^i + x_t U^i + h_{t-1} V^i)$$

$$h = o * tanh(C_t)$$

5.4. Programska implementacija

Arhitektura LSTM mreže korištena u implementaciji sustava za automatsko raspoznavanje govora napravljena je po shemi prikazanoj na slici 25.



Slika 25. Arhitektura LSTM mreže

Horizontalna komponenta predstavlja broj vremenskih koraka, a ulazni podaci dani su dvama slojevima LSTM mreže. Kao ulaz mreži dani su MFCC koeficijenti ulaznog zvučnog signala, a izlaz se sastoji od 11 vrijednosti koje predstavljaju vjerojatnosnu distribuciju za svaku od mogućih znamenki te dodatno za klase „*unknown*“ i „*silence*“.

Izlazne vrijednosti drugog LSTM sloja prethodno su dane potpuno povezanom sloju zvanom „*Time Distributed*“ sa softmax aktivacijskom funkcijom, koji se koristi kod povratnih neuronskih mreža

5.4.1. Izrada modela

Za izradu modela korištena je biblioteka Keras pisana u programskom jeziku Python. Keras korisniku pruža API visoke razine koji se u pozadini oslanja na TensorFlow, Theano ili neki drugi radni okvir te pruža implementaciju brojnih alata korištenih u aspektima strojnog učenja, između ostalog i podršku za konvolucijske i povratne neuronske mreže.

```
hidden_length = 100

model = Sequential()

model.add(Embedding(1000, 100, input_length = X.shape[1]))
model.add(LSTM(hidden_length))
model.add(LSTM(hidden_length))

if use_dropout:
    model.add(Dropout(0.5))

model.add(TimeDistributed(Dense(11)))
model.add(Activation('softmax'))

model.build(loss = 'categorical_crossentropy', optimizer='adam', metrics = ['accuracy'])
```

Slika 26. Izrada LSTM slojeva

Isječak koda kojim se izrađuje opisana LSTM mreža prikazan je slikom 26. Dodatno, između drugog LSTM i potpuno povezanog sloja dodan je i „Dropout“ sloj s vjerojatnošću od 20%, što znači da će jedan od pet primjera biti nasumično isključen prilikom svakog ažuriranja težina. „Dropout“ je jedan od načina regularizacije modela te pridonosi boljoj generalizaciji i robusnosti.

5.4.2. Točnost modela

S navedenom arhitekturom provedeno je dvije vrste testova. Jedna vrsta testa bila je treniranje i testiranje na skupu podataka samo jednog izvornog govornika, a druga je ona generalnija, trenirana i testirana na cijelom skupu podataka gdje se nalaze zvučni zapisi više govornika različitih dobnih skupina i spola.

Točnost modela izračunana je kao postotak uspješno klasificiranih zvučnih signala u testnome skupu.

Govornik	Točnost modela / %	Točnost modela iz [1] / %
Muški	78.96	91.66
Ženski	81.22	88.00

Tablica 2. Rezultati modela za prepoznavanje glasa na istim govornicima

Tablica 2 prikazuje točnost modela treniranih i testiranih na jednom muškom govorniku te na jednom ženskom govorniku. treći stupac prikazuje rezultate modela baziranog na povratnoj neuronskoj mreži implementiranog u radu [1].

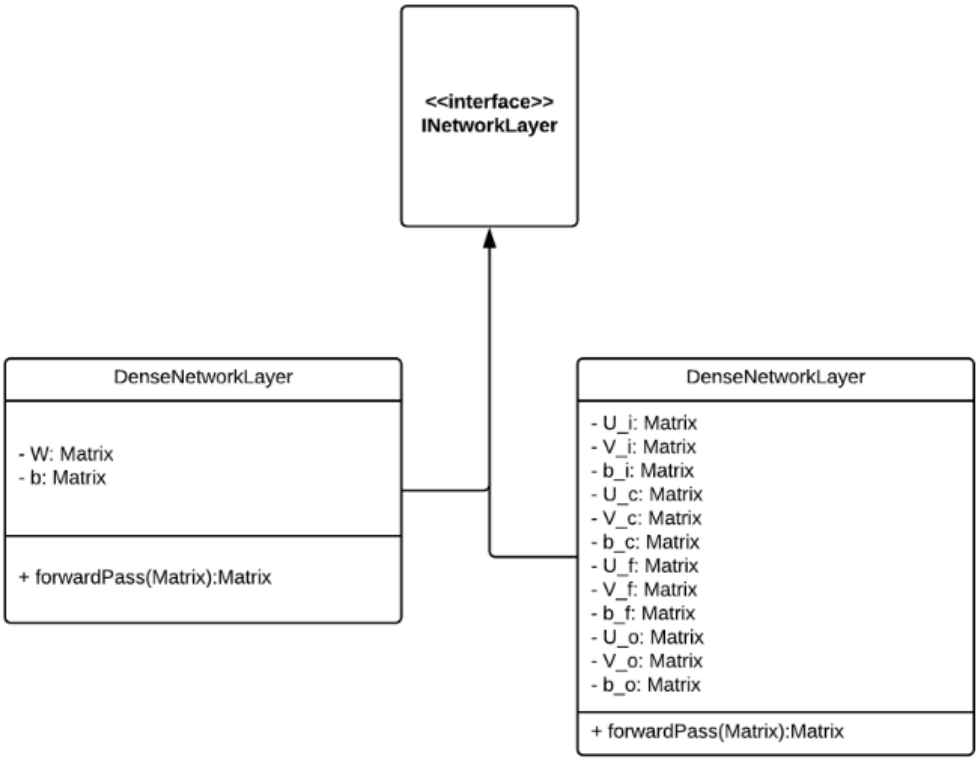
Točnost modela / %	Točnost modela iz [2]
72.50	98.9 – 99.5

Tablica 3. Rezultati modela za prepoznavanje glasa na različitim govornicima

Tablica 3 prikazuje točnost modela treniranih i testiranih na različitim govornicima te usporedbu rezultata modela baziranog na LSTM povratnoj neuronskoj mreži iz rada [2].

5.4.3. Učitavanje modela

U sklopu ovoga rada napravljena je i implementacija programa u programskom jeziku Java koja učitava gotovi model iz datoteke dobivene nakon treniranja pomoću biblioteke Keras u pogramskom jeziku Python.



Slika 27. UML dijagram hijerarhije slojeva mreže

Na slici 27 prikazan je UML dijagram hijerahije slojeva korištenih izvedbi. Sučelje „INetworkLayer“ predstavlja apstraktni sloj neuronske mreže, a „DenseNetworkLayer“ i „LSTMNetworkLayer“ predstavljaju konkretne implementacije potpunog, odnosno LSTM sloja.

6. Suparnički primjeri

Suparnički primjeri ulazi su u modele strojnog učenja koje je napadač namjerno dizajnirao u svrhu da model napravi pogrešku, odnosno da ulaz klasificira kao bilo koju drugu proizvoljnu klasu. Ovakvi primjeri predstavljaju veliki udar na modele strojnog učenja te predstavljaju značajan problem zbog kojega je potrebno uložiti ozbiljan trud u istraživanju ovoga područja.

Suparnički primjeri mogu biti jako opasni. Na primjer, napadač može napasti autonomna vozila koristeći vizualne alate (naljepnice, boja) kako bi napravio suparnički „stop“ znak koje bi vozilo moglo interpretiralo kao „prednost“ ili neki drugi znak.

6.1. Definicija

Suparnički primjeri mogu se matematički formalizirati kao ulazni podatak **x'** za kojega vrijedi sljedeća formula:

$$f_\theta(x') \neq y$$

odnosno odluka modela koji se može predstaviti funkcijom **f** s parametrima theta, nije prava oznaka ulaznog podatka. Također, na suparnički se primjer postavlja i zahtjev koliko može odstupati od pravog primjera veličinom eps:

$$||x' - x||_p < \epsilon$$

Formula (2) udaljenost između suparničkog i originalnog primjera računa određenom normom, a neke od mogućnosti su:

- L0 – ograničava broj elemenata vektora koji mogu biti mijenjani u x' u odnosu na vektor x
- L2 – ograničava kvadratnu udaljenost između odgovarajućih elemenata vektora x' i x
- $L_{infinite}$ – ograničava maksimalnu udaljenost između odgovarajućih elemenata vektora x' i x

6.2. Kategorizacija suparničkih primjera

U nastavku su prikazane moguće kategorizacije napada suparničkim primjerima. Napadi se mogu podijeliti na temelju suparničke falsifikacije, znanja napadača, suparniče specifičnosti te frekvenciji napada.

6.2.1. Suparnička falsifikacija

Suparnička falsifikacija (engl. *Adversial falsification*).

- „False positive“ – generiraju negativne primjerke koji su krivo klasificirani kao pozitivni, na primjer u sustavu za prepoznavanje zloćudnih programa, ako se benigni program klasificira kao zloćudni, to se označava kao „false positive“. U sustavu za prepoznavanje govora to može biti suparnički zvučni signal, koji je neprepoznatljiv za čovjeka, dok ga model duboke neuronske mreže klasificira u neku klasu s velikom pouzdanošću.
- „False negative“ – ova je pogreška najzastupljenija u suparničkim primjerima, gdje se dogodi situacija da čovjek može klasificirati zvučni signal, ali neuronska mreža ne može.

6.2.2. Znanje napadača

Napadi koje napadači provode nad modelima strojnog učenja mogu se podijeliti u dvije kategorije na temelju znanja napadača (engl. *Adversarys knowledge)*:

- „White box“ napadi – napadač ima pristup cijeloj arhitekturi modela, uključujući i podacima za treniranje, gradijentima, hiperparametrima, aktivacijskim funkcijama, broju slojeva itd. Većina napada na modele strojnog učenja jesu „White box“ napadi.
- „Black box“ napadi – napadač nema pristup unutarnjoj strukturi niti gradijentima, samo konačnoj odluci modela.

6.2.3. Suparnička specifičnost

Suparnička specifičnost (engl. *Adversial specificity*).

- Ciljano (engl. *targeted*) – model duboke neuronske mreže se navodi na klasifikaciju u točno određenu klasu. Na primjer, u sustavu koji koristi

biometriju, odnosno autorizaciju korisnikovog otiska prsta ili lica, napadač se želi predstaviti kao točno određeni korisnik.

- Ne ciljano (engl. *Non targeted*) – ne pridjeljuju točno određenu klasu izlazu neuronske mreže, klasa može biti proizvoljna, bilo koja osim originalne. Ovakvi su napadi jednostavniji za implementirati nego ciljani napadi s obzirom da postoji više opcija i veći prostor na kojeg se izlaz može preusmjeriti

6.2.4. Frekvencija napada

Frekvencija napada (engl. *Attack frequency*)

- Jednokratni napadi (engl. *One time attacks*) – potreban je samo jedan pokušaj kako bi se optimizirao suparnički primjer.
- Iterativni napadi (engl. *Iterative attacks*) – potrebno je više pokušaja kako bi se optimizirao suparnički primjer.

Uspoređujući ih, iterativni napadi obično daju bolje suparničke primjere, ali zahtijevaju više interakcije (upita) s modelom koji se napada te samim time i više računalnog vremena i resursa kako bi ih se generiralo.

6.3. Metoda “Fast Gradient Sign”

Ova se metoda pojavila 2014. te zapravo predstavlja linearne petrubacije koje dodaju jako male vektore ulaznom signalu, koji sadrže elemente jednake predznaku elemenata gradijenta funkcije gubitka s obzirom na ulazni podatak. Postupak se može prikazati sljedećom formulom:

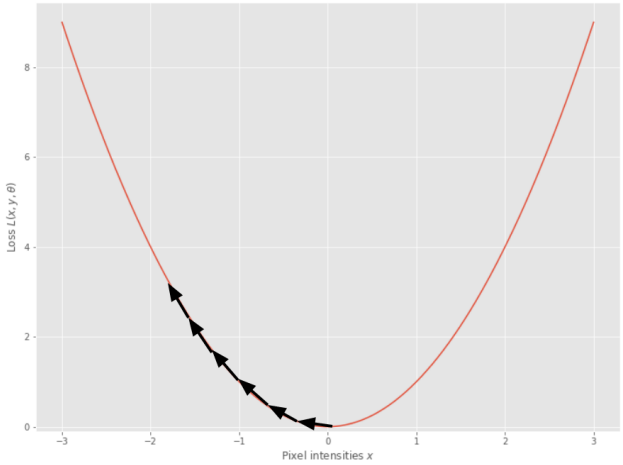
$$\omega = \epsilon \, sign(\Delta J(\theta, x, y))$$

$\boldsymbol{\theta}$ predstavlja parametre modela, **x** je ulazni podatak, a **y** odredišna klasa. Ovaj postupak se odrađuje određen broj iteracija, sve dok se ne dobije zvučni signal koji se klasificira kao željena klasa, a nije osjetno promijenjen. Jedan korak iteracije može se prikazati formulom:

$$x = x - \omega$$

Vjerojatno je došlo do nekakve pogreške u implementaciji s obzirom da niti za jedan signal nije dobiven uspješni rezultat. Testirane je provedeno na

nekoliko signala različitih govornika, ali bi se u svakom slučaju došlo do prevelikog broja iteracija koje bi neprepoznatljivo izobličile signal.



Slika 28. Grafički prikaz FGS metode

7. ZAKLJUČAK

Cilj ovoga rada bila je izrada sustava za automatsko prepoznavanje glasa, kako je to prezentirano u poglavlju 5. Dobiveni rezultati u prvome testu nešto su lošiji nego oni iz rada baziranog na povratnim neuronskim mrežama, ali u teoriji LSTM neuronske mreže bi trebale pokazivati puno bolje rezultate nego RNN. Drugi test pokazao je znatno lošije rezultate neki referentna izvedba. Razlozi lošijim rezultatima mogu biti razlike u korištenim skupovima podataka, neke različitost u implementaciji i regularizaciji modela. Također, u poglavlju 6 predstavljeni su suparnički primjeri te njihov pokušaj implementacije.

Budući radi bi se mogao bazirati na poboljšanju metode generiranja suparničkih primjera, te bi se primjeri mogli testirati na nekim od postojećih sustava za raspoznavanje glasa. Također, postoji i mogućnost analize i implementacije neki drugih metoda za generirane suparničkih primjera, ali i nekih drugih napada na modele strojnog učenja. U ovome radu ostavljen je i prostor za doradu modela za raspoznavanje glasa s metodama za obranu od ovakvih napada.

8. Literatura

1. R.L.K. Venkateswarlu, V. Kumari, G.V. Jayasri, „Speech Recognition By Using Recurrent Neural Networks“, lipanj 2011
2. A. Graves, D. Eck, N. Beringer, J. Schmidhuber, „Biologically Plausible Speech Recognition with LSTM Neural Nets“, 2004.
3. N. Carlini, D.Wagren, „Audio Adversial Examples: Targeted Attacks on Speech-To-Text“, Berkley 2018
4. D. Iter, J. Huang, M. Jermann, „Generating Adversial Examples for Speech Recognition“, Stanford
5. X. Tian et al, „Deep LSTM For Large Vocabulary Continuous Speech Recognition“, Beijing, China, 2017
6. X. Yuan et al, „Adversial Examples: Attacks and Defenses for Deep Learning“, University of Florida, 2018
7. N. Papernot, P. McDaniel, „Crafting Adversial Input Sequences for Recurrent Neural Networks“, 2016
8. Keras, s Interneta, https://www.keras.io, 21 prosinac 2018
9. MFCC, s Interneta, https://en.wikipedia.org/wiki/Mel-frequency_cepstrum 18 studeni 2018
10. G. H. Patel College of Engineering, „Feature Extraction Methods LPC, PLP and MFCC In Speech recognition“, India, 2013