SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1394

RASPOZNAVANJE IZGOVORENOG JEZIKA IZ KRATKOG ZVUČNOG ZAPISA METODAMA STROJNOG UČENJA

Lana Bartolović

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1394

RASPOZNAVANJE IZGOVORENOG JEZIKA IZ KRATKOG ZVUČNOG ZAPISA METODAMA STROJNOG UČENJA

Lana Bartolović

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1394

Pristupnica:	**Lana Bartolović (0036538315)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	izv. prof. dr. sc. Alan Jović

Zadatak: **Raspoznavanje izgovorenog jezika iz kratkog zvučnog zapisa metodama strojnog učenja**

Opis zadatka:

Raspoznavanje izgovorenog jezika iz kratkog zvučnog zapisa važan je preduvjet za ostvarenje izgradnje učinkovitih višejezičnih računalnih prevoditelja u stvarnom vremenu. Cilj ovog završnog rada je izrada klasifikacijskog modela koji će na temelju kratkog zvučnog zapisa (do 20 sekundi) izgovorenog jezika na nekom od svjetskih jezika ustanoviti o kojem se jeziku radi. Kao skup za učenje modela potrebno je koristiti neki od dostupnih skupova podataka zvučnih zapisa na svjetskim jezicima na internetu, kao što je Spoken Language Identification, dostupan na web sjedištu Kaggle (https://www.kaggle.com/datasets/toponowicz/spoken-language-identification). Klasifikacijski model treba biti zasnovan na nekom od algoritama strojnog učenja, uključujući algoritme dubokog učenja. Primjeri takvih algoritama su konvolucijske neuronske mreže, povratne neuronske mreže i transformerske arhitekture. U radu je također potrebno razmotriti metode predobrade zvučnih zapisa snimljenog govora kako bi se poboljšala učinkovitost izgrađenog modela. Implementaciju je potrebno napraviti u programskom jeziku po vlastitom izboru, a za izgradnju modela može se u slučaju nedostatka vlastitih sklopovskih resursa koristiti dostupna web rješenja (npr. Google Colab).

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

1. Uvod

Digitalizacija današnjeg svijeta dovodi do sve veće potrebe komunikacije ljudi i računala. Ne ostavlja dojam teškog problema, ali prevesti ljudski jezik u ono s čime računalo može raditi i dalje je neriješen, odnosno neusavršen problem u znanosti. To je otvorilo vrata u zasebno područje znanosti usmjereno na olakšavanje tog izazovnog zadatka – obrada prirodnog jezika (engl. *natural language processing*, NLP). NLP je danas vrlo važno područje zbog rastućeg broja primjena, stoga je ovaj rad usmjeren na jedan od ključnih i fundamentalnih zadataka: određivanje izgovorenog jezika. Ideja je moći koristiti razvijeni algoritam u nekim od pravih problema, poput implementacije algoritma u mobilnoj aplikaciji kako bi poboljšao korisnikovo iskustvo ili pak iskoristiti model kao dio nekog većeg jezičnog modela.

Cilj ovog završnog rada jest razviti algoritam umjetne inteligencije koji bi omogućio klasifikaciju izgovorenog jezika iz audio zapisa. Algoritam je učen klasificirati između tri jezika: engleski, španjolski i njemački. U radu se razvijaju i uspoređuju tri zasebna modela s ciljem dobivanja što boljih rezultata.

U nastavku je ukratko opisano područje obrade prirodnog jezika. Dane su informacije o korištenom skupu podataka. Kasnije je dan detaljan pregled metoda dubokog učenja korištenih u razvoju modela. Rezultati su izneseni na kraju rada.

2. Obrada prirodnog jezika

Obrada prirodnog jezika je područje umjetne inteligencije fokusirano na omogućavanje računalu razumijevanje i manipulaciju ljudskim jezikom [1]. NLP svoje korijene ima još u 1940-ima, ali tada nije bilo smatrano zasebnom granom znanosti. Nakon Drugog svjetskog rata ljudi su shvatili važnost prevođenja s jednog jezika na drugi te su htjeli razviti stroj koji bi to činio za njih. Pokušaji rješavanja ovog naizgled jednostavnog problema prošli su kroz više drastično različitih faza; na početku se tvrdilo da će se "problem strojnog prevođenja riješiti u narednih tri do pet godina" [1], takav veliki optimizam 10-ak godina kasnije bio je popraćen rezanjem novčanih sredstava uslijed neuspjeha. Od područja se praktički odustalo sve dok se 1980-ih nisu krenuli razvijati tzv. ekspertni sustavi koji su opet, puno obećavali, ali se i od njih ubrzo odustalo jer nisu ispunili očekivanja. Nastupio je period tzv. AI zime, gdje je opet palo zanimanje za umjetnom inteligencijom. U početku rezultati nisu bili zadovoljavajući jer su znanstvenici problemu pristupali na loš način. Svi su pokušaju bili simboličke prirode (pristupanje znanju kao simbolima te manipulacijom nad njima putem definiranih pravila). Taj je pristup bio zamijenjen statističkim pristupom koji je pokazivao bolje rezultate, ali glavna mana mu je bila vrlo zahtjevna ekstrakcija značajki.

Teško je shvatiti zašto NLP i dalje nije u potpunosti riješeno područje u znanosti zato što su problemi kojima se NLP bavi kod ljudi riješeni spontano, odnosno ljudi ni sami nisu svjesni kako rješavaju takve probleme, poput određivanja subjekta u rečenici ili pak detekcije sentimenta u tekstu. Kod ljudi ne postoji algoritam u glavi koji rečenicu pretvara u niz simbola iz kojih se pravilima logike pokušava izvesti zaključak, što je između ostalog i dovelo do odustanka od tradicionalnog, simbolističkog pristupa i traženja novijeg, sličnijeg ljudskom razmišljanju. Tako je razvijen NLP zasnovan na neuronskim mrežama, a taj je pristup dominantan i danas. Ono što omogućava uspjeh takvog NLP-a

je ogromna količina dostupnih podataka razvojem Interneta. Možemo povući paralelu s ljudskim učenjem i iskustvom: svatko će imati više samopouzdanja u rješavanje proizvoljnog zadatka i točnije će ga riješiti ako je prije imao priliku susresti se s njime ili nečim bliskim. Takva je i situacija s neuronskim mrežama jer se tu događa indukcija: na osnovu već viđenih primjera izvodi se generalno pravilo koje se onda primjenjuje na novim primjerima. Nije teško zaključiti da se generalno uz veći skup podataka za učenje postižu bolji rezultati [2].

Vidimo kako su u današnjem svijetu životi ljudi isprepleteni s postojanjem računala te je logična posljedica ove činjenice potreba za razvojem NLP-a. Neke od najčešćih primjena NLP-a su [1, 3]:

1. Prepoznavanje govora

Iz zvučnog zapisa generira se njegov tekstualni zapis.

2. Generiranje zvučnog zapisa teksta (engl. *text to speech*)

Ovo je suprotno od prve točke. Na temelju teksta potrebno je generirati zvučni zapis koji dosljedno reprezentira tekst. Primjene su razne, npr. za pomoć osobama s oštećenjem vida.

3. Prevođenje jezika

4. Filtriranje e-mailova

Filtriranje primljenih e-mailova po naučenom korisnikovom ukusu.

5. Analiza sentimenta u tekstu

Za dani ulazni tekst potrebno je odrediti u koju kategoriju pripada, npr. tužan, sretan, ljutit, prestrašen i sl. Može se primjenjivati na recenzijama proizvoda.

6. Pametni asistenti

Pametni asistenti ubrzavaju uporabu tehnoloških sprava tako što uz pomoć NLP-a razumiju ljudski govor i pomažu u obavljanju zadataka, kao što su postavljanje alarma, pronalazak restorana u blizini i sl.

Kao što je već rečeno, problemi kojima se NLP bavi su vrlo teški te se moraju razmatrati u više koraka. Koraci obično prate tzv. cjevovod za NLP, a ključne točke su [4]

1. Prikupljanje podataka

Dobar skup podataka je vrlo važan element bilo kakvog algoritma zasnovanog na neuronskim mrežama i značajno utječe na rezultate.

2. Predobrada skupa podataka

U ovo se ubrajaju različiti postupci za čišćenje skupa podataka i njegova priprema za daljnju uporabu.

3. Ekstrakcija značajki

Ovdje se događa prvo usklađivanje čovjeka i računala – pretvorba riječi u numeričke značajke s kojima računalo zna baratati. Ima mnogih tehnika, kao što je *One-Hot Encoding* – kodiranje riječi binarnim vektorima.

4. Modeliranje

Srž cjevovoda za NLP, ovdje se događa učenje. Postoje različite tehnike, a u ovom radu je korišten pristup temeljen na dubokom učenju, vidjeti poglavlje 4.

5. Evaluacija

Nakon što smo završili učenje, u fazi evaluacije vrednujemo dobiveni rezultat.

6. Uvođenje

Ovdje se vraća na ono što nam je cijelo vrijeme i bio cilj: omogućiti uporabu algoritma u stvarnim situacijama (razne digitalne aplikacije, drugi modeli...).

Ovaj rad bavi se jednim dijelom koraka predobrade, a to je detekcija jezika. Ključno je prepoznati o kojem se jeziku radi kako bi se kasnije mogli primijeniti alati specifičniji za taj jezik i dobiti bolji rezultati. To da jednojezični modeli nadmašuju višejezične modele su pokazali znanstvenici u radu [5]. U skladu s ovim rezultatima, u ovom se radu provodi klasifikacija izgovorenog jezika iz zvučnog zapisa u jednu od tri klase: engleski

jezik, španjolski jezik te njemački jezik. Model kao takav bi se mogao koristiti kao korak predobrade, prije davanja ulaznih vrijednosti jednojezičnom modelu.

3. Skup podataka

3.1. Važnost skupa podataka

Vrlo važan i obavezan korak u bilo kakvom algoritmu temeljenom na strojnom učenju jest prikupljanje podataka. Podataka danas ima u izobilju, ali je izazov pronaći dovoljno kvalitetnih podataka, specifičnih za dani problem. Kvaliteta podataka je ta koja može značajno pomoći u postizanju dobrih rezultata. Zbog čega je tako? Budući da algoritam mora učiti na podacima koji su prikupljeni, oni moraju točno reprezentirati područje istraživanja. Algoritam sam za sebe ne zna ništa i sve što će naučiti bit će temeljeno na podacima koji su prikupljeni, stoga ako se koriste loši podaci, algoritam će se pokušati njima prilagoditi te će izvući neželjene zaključke. Primjeri loših karakteristika skupova podataka su krivo označeni podaci (ako se za plavu boju kaže da je crvena, ne može se očekivati da će algoritam za primjere s plavom bojom dati točan rezultat), nedovoljan broj podataka (ako je skup podataka premalen, a problem pretežak, nećemo dobiti preciznu reprezentaciju problema te će algoritam za nove, neviđene primjere davati jako loše rezultate), neuravnotežen skup podataka (npr. ako se problem tiče svrstavanja slika u jednu od dvije klase, a skup podataka sadrži 90% podataka koji pripadaju prvoj klasi, algoritam će za dane primjere iz druge klase davati loše rezultate) [6].

3.2. Podaci korišteni u ovom radu

Postoje mnogi skupovi podataka koji bi se mogli iskoristiti za ovaj problem, poput Mozilla common voice [7]. Taj skup podataka sadrži puno podataka što je i poželjno, ali je obrada takvog skupa podataka jako skupa. Stoga je skup podataka za ovaj rad uzet s javno dostupne web stranice Kaggle [8]. Korištena su dva skupa podataka.

3.2.1. Skup podataka Spoken Language Identification

Sastoji se od zvučnih zapisa na engleskom, španjolskom i njemačkom jeziku [9]. Ovaj skup podataka inspiriran je natjecanjem Spoken Languages 2 [10], kao poboljšanje za taj skup podataka jer je imao nedostataka (najčešće samo jedan muški govornik za svaki od 176 jezika, a isti govornici sudjelovali su i u podacima za učenje i testiranje, vidjeti poglavlje 4.2.1.) Tijekom predobrade podataka radi se augmentacija. Augmentacija je proces kojim se skup podataka umjetno povećava tako što se nad prikupljenim podacima rade male promjene, poput dodavanja šuma, kod slika su mogući rotiranje, translacije i slično. Promjene su bile napravljene ili nad brzinom zapisa ili nad visinom glasa u zapisu ili je zapisu dodan šum, poput prolaska auta. Tako je od početnog skupa podataka dobiven novi koji sadrži 73080 zvučnih zapisa koji su korišteni za učenje, a 540 za validaciju. Zapisi su spremljeni u .FLAC (engl. *Free Lossless Audio Codec*) formatu kako bi se omogućilo efikasno spremanje zapisa bez gubitka njegove kvalitete. Svaki je zapis u obliku

(language)_(gender)_(recording ID).fragment(index)[.(transformation)(index)].flac

gdje je brzina u rasponu 1–8, visina u rasponu 1–8 te šum u rasponu 1–12. Uvijek se primjenjuje samo jedna od tri promjene. Podaci su podjednako raspoređeni između jezika, spola i govornika što je bitno za kasniju obradu.

3.2.2. Skup podataka Spoken Languages

Budući da je skup podataka Spoken Language Identification podijeljen na dva dijela, a za algoritam su potrebne tri nezavisne grupe, kao treća grupa uzet je novi skup podataka [11]. Ovakav pristup može biti problematičan jer nema garancije da su dva različita skupa podataka jednako kvalitetna i to može značajno utjecati na rezultate. Međutim, uz prikladnu pripremu podataka taj rizik se smanjuje i podaci se mogu koristiti. Dapače, korištenje različitog skupa podataka za testiranje bolje reprezentira uzorke iz stvarnog svijeta i može dati točniju procjenu [12]. Ovaj skup podataka sadrži ukupno 510 podataka, 170 za svaki jezik. Razlike u odnosu na prvi skup podataka su te da su ovi podaci u .wav formatu i nema augmentacije podataka. Zapisi su također duljine 10 sekundi.

4. Duboko učenje

Kako bi se objasnio pojam dubokog učenja, potrebno je prvo reći ponešto o strojnom učenju.

4.1. Strojno učenje

Strojno učenje je dio umjetne inteligencije koje se bavi razvojem i učenjem statističkih algoritama koji uče na podacima te to primjenjuju na novim zadacima, odnosno mogu obavljati zadatke bez eksplicitnih uputa [13]. Drugim riječima, strojno učenje koristi se za ekstrakciju informacija iz podataka i njihovu pohranu tako da se kasnije mogu koristiti [14]. Algoritam strojnog učenja, kad se primijeni na ulazne podatke, rezultira modelom. Model je na početku definiran do na neku razinu, a uči tako da optimira kriterij uspješnosti (npr. ukupnu grešku) na temelju podataka [15]. Postoji više pristupa strojnom učenju: nadzirano, nenadzirano i podržano učenje. U ovom radu korišten je model nadziranog strojnog učenja te je sukladno tome nadzirano učenje opisano u nastavku.

4.1.1. Nadzirano strojno učenje

Naziv nadzirano učenje odnosi se na podatke na kojima model uči. Kod nadziranog učenja, podaci su grupirani kao parovi (x, y) = (ulaz, labela) gdje je labela očekivani izlaz modela za dani ulaz. Zadaća modela je naći preslikavanje

$$\hat{y} = f(x) \tag{4.1}$$

Primjer je model koji uči prepoznati imenice; kod njega će ulazni podaci biti u obliku npr. (jabuka, 1), (suknja, 1), (gledati, 0), gdje 1 predstavlja klasu "imenica", a 0 predstavlja klasu "nije imenica". Prethodno opisan primjer spadao bi u problem klasifikacije, gdje

se ulazne primjere pokušava svrstati u prethodno definirane klase. Ako postoje dvije klase, govorimo o binarnoj klasifikaciji, u suprotnom o višeklasnoj klasifikaciji. Takav je i problem kojim se bavi ovaj rad; ulaz treba svrstati u jednu od tri klase.

4.1.2. Algoritmi strojnog učenja

Algoritam strojnog učenja je procedura koja se vrši nad skupom podataka i u konačnici rezultira modelom. To je skup pravila koji definiraju način učenja iz podataka i samu strukturu modela. Postoji više algoritama koji se mogu primijeniti od kojih su neki od poznatijih algoritama za nadzirano učenje stabla odluke, Bayesov klasifikator i umjetne neuronske mreže. Kako je u ovom radu korišten algoritam umjetnih neuronskih mreža, taj je opisan u nastavku.

Umjetne neuronske mreže

Umjetne neuronske mreže (engl. *Artificial neural networks*, ANN) su algoritam i način predstavljanja znanja zasnovan na ljudskom mozgu. Biološki neuroni su temeljna gradivna jedinica živčanog sustava i omogućavaju primanje signala iz vanjskog svijeta, razmišljanje, bilo kakve motoričke akcije; ukratko svu esencijalnu funkcionalnost ljudi. Svaki je neuron povezan s prosječno 7000 drugih neurona [16]. Sastoje se od tijela (some), dendrita, aksona i završnih članaka, slika 4.1. Tijelo neurona je dio gdje se obavlja glavna funkcionalnost neurona. Akson je dio neurona koji služi prijenosu signala do drugih neurona, a završni članci su dio na samom kraju aksona gdje se događa otpuštanje neurotrasmitera (molekule koja je srž komunikacije dva neurona [17]). Dendriti su mjesto gdje neuron prima signale od drugih neurona s kojima je povezan. Na temelju ovih neurona, uvedeni su umjetni neuroni koji oponašaju ovu funkciju. Ukratko, svaki umjetni neuron prima informaciju od određenog broja neurona, a određenom broju neurona on šalje svoju informaciju. Naravno, to nije tako jednostavno i postoji više vrsta ANN-ova. Arhitektura ANN-a govori kako su neuroni u mreži povezani i koliko ih ima. Neuronska mreža sastoji se od skupina neurona: slojeva. Mreža može biti unaprijedna, gdje svaki neuron prima informacije isključivo od sloja neposredno prije, slika 4.3., rezidualna gdje svaki neuron prima informacije isključivo od slojeva prije ili rekurentna gdje neuroni mogu primati informacije od neurona s proizvoljne lokacije. Svaki neuron prima informacije u obliku težinske sume izlaza drugih slojeva, slika 4.2., a kao izlaz (osim neurona

u izlaznom sloju) vraća rezultat djelovanja aktivacijske funkcije na težinsku sumu. Neke od često korištenih aktivacijskih funkcija su

zglobnica (ReLU)

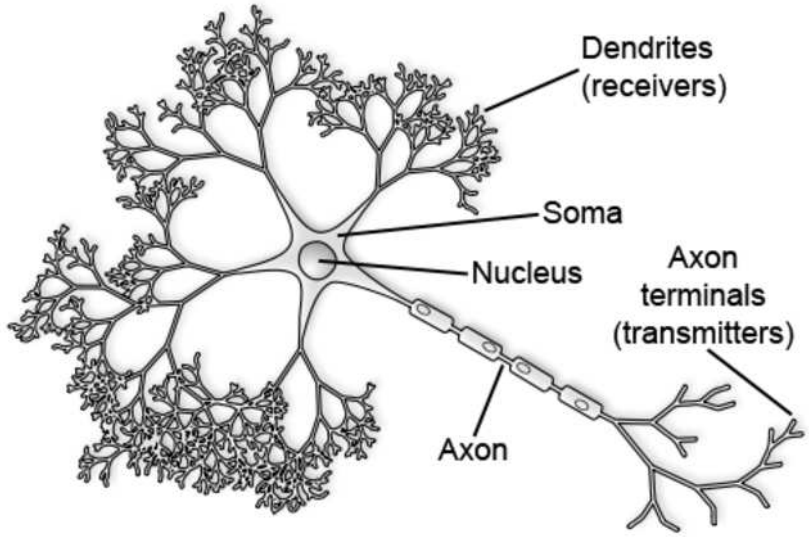
$$\mathrm{ReLU}(x) = \max(0, x) \tag{4.2}$$

sigmoidalna funkcija

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.3}$$

i tangens hiperbolni

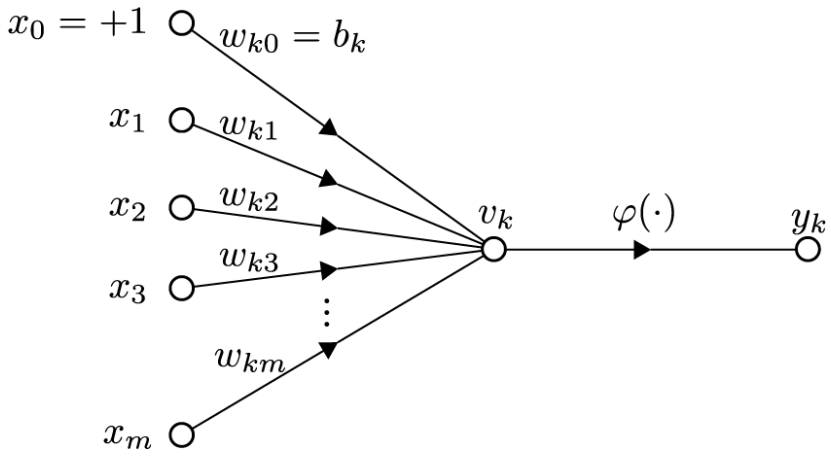
$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4.4}$$



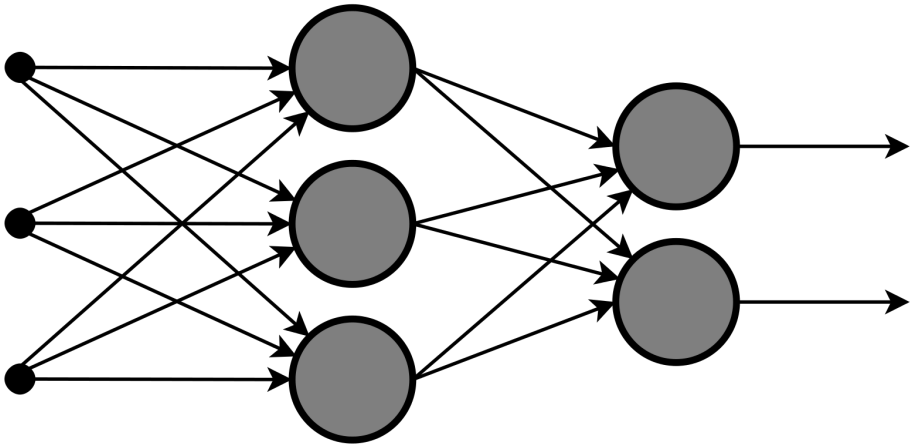
Slika 4.1. By Nicolas.Rougier - File:Neuron-figure-notext.svg, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=54107167

Za aktivacijsku funkciju je bitno da ne bude linearna, jer se u tom slučaju izlaz iz mreže može napisati kao težinska suma ulaza i mreža se ponaša kao da ima samo jedan neuron, a to je nepoželjno jer takve mreže pokazuju puno manju ekspresivnost i sposobnost generalizacije naspram većima. Izlazni sloj je mjesto gdje se događaju predikcije, a u problemu klasifikacije broj neurona izlaznog sloja odgovara broju klasa. Težinske sume izlaznih neurona najčešće se provlače kroz funkciju softmax koja izlazne vrijednosti adekvatno preslikava na interval $[0, 1]$ tako da one reprezentiraju vjerojatnosti pripadanja toj klasi. Konačni izlaz iz modela odgovara klasi s najvećom vjerojatnosti.

Sad kad su objašnjene osnove teorije, slijedi dio o dubokom učenju.



Slika 4.2. "Diagram of artificial neuron" by Mikadukimooo, adapted from the original GIF, licensed under CC BY-SA 4.0. Available at: `https://commons.wikimedia.org/wiki/File:Artificial_neuron-2.gif`



Slika 4.3. "A directed graph representation of an artificial feed-forward neural network" by Offnfopt, licensed under CC BY-SA 3.0. Available at: `https://commons.wikimedia.org/wiki/File:Multi-Layer_Neural_Network-Vector-Blank.svg`

4.2. Duboko učenje

Duboko učenje je potpodručje strojnog učenja zasnovano na dubokim neuronskim mrežama, odnosno mrežama s mjerom CAP (engl. *credit assignment path*) većom od 2. Kod unaprijednih mreža taj je broj jednak broju skrivenih slojeva + izlazni sloj, inače je potencijalno neograničen [18]. Primjene dubokog učenja su mnoge, kao na primjer u području računalnog vida, autonomna vozila pa i NLP.

Zbog čega je duboko učenje važno? U današnjem svijetu, podataka ima ekstremno mnogo. Kao što je već rečeno, učenje na većem skupu podataka u pravilu vodi do boljih rezultata. Duboke neuronske mreže su omogućile obradu velike količine podataka i zadataka veće kompleksnosti, a uz to pružaju veću točnost [18,19]. Ovi su rezultati veći-

nom zato što duboke neuronske mreže imaju kompleksniju arhitekturu i mogu uhvatiti preciznije značajke. No, veća kompleksnost nosi i loše stvari, primarno prenaučenost i dugo vrijeme obrade. Prenaučenost je problem koji se očituje u tome da model jako dobro nauči primjere iz skupa podataka, ali se na neviđenim podacima ponaša značajno lošije jer se previše prilagodio skupu podataka. Kod rada s dubokim neuronskim mrežama potrebno je pronaći optimum između dovoljno velike kompleksnosti za obavljanje zadatka i dovoljno male kompleksnosti za izbjegavanje prenaučenosti. Jedan od načina sprečavanja prenaučenosti je metoda unakrsne provjere.

4.2.1. Metoda unakrsne provjere

Kod rada s modelima, jedini način da se dobije nepristrana procjena kvalitete modela jest provjera rada modela na potpuno novim podacima. To se postiže podjelom skupa podataka na skup za učenje i skup za testiranje, gdje se podaci iz skupa za učenje koriste za učenje modela, a skup za testiranje glumi neviđene primjere za evaluaciju modela. Kako bi se pokušala izbjeći prenaučenost, još se radi i metoda unakrsne provjere. To je zapravo daljnja podjela skupa za učenje na skup za učenje i skup za provjeru. Skup podataka za provjeru se koristi u procesu učenja, kao uvid u to kako bi se model ponašao na neviđenom skupu podataka; za sprečavanje prenaučenosti modela na primjere iz skupa za učenje tako što se u epohama učenja, vidjeti poglavlje 4.2.2., evaluira ponašanje modela na skupu za provjeru, kao na neviđenim podacima, i kad pogreška na skupu za provjeru poraste, to upućuje na prenaučenost. Funkcija skupa za učenje ostaje ista [15].

Originalan skup podataka najčešće se dijeli u omjeru 60-80% za učenje, 10-20% za testiranje i 10-20% za provjeru [20]. Jako je bitno da ne dolazi do curenja informacija (npr. da se isti podaci nalaze i u skupu za učenje i u skupu za testiranje) između ovih skupova podataka jer to vodi do neispravne procjene modela.

4.2.2. Učenje

Učenje duboke neuronske mreže svodi se na promjenu jakosti veza između neurona s ciljem smanjenja ukupne greške. Ukupna greška je funkcija koja modelira odstupanje predviđene vrijednosti od prave vrijednosti. Kod klasifikacije, to je funkcija gubitka (engl. *loss function*). U ovom radu korištena je funkcija gubitka kategorička unakrsna entro-

pija za rijetke podatke (engl. *sparse categorical cross-entropy loss function*) koja traži da labele klasa ulaznih primjera budu cijeli brojevi. Izračun za pojedini primjer je iskazan formulom

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\log(p_{i,y_i}) \tag{4.5}$$

gdje je N broj obrađenih uzoraka, y_i labela prave klase i-tog primjerka, a p_{i,y_i} vjerojatnost koju je model dao kao rezultat za klasu y_i (podsjećam, vjerojatnost pripadanja primjerka toj klasi). Za smanjenje ukupne greške može se koristiti gradijent funkcije. Gradijent funkcije govori u kojem je smjeru najveći rast funkcije. U skladu s time, u smjeru suprotnom od gradijenta funkcija najviše pada. Budući da se radi o funkciji greške, nju se želi smanjiti te se zato traži gradijent da bi se poboljšali parametre mreže (težine). Računanje gradijenta svodi se na računanje parcijalnih derivacija funkcije po njenim parametrima. Budući da je to vremenski zahtjevno jer mreže imaju velik broj parametara, koristi se postupak propagacije pogreške unatrag koji za parcijalnu derivaciju funkcije greške po elementu jednog sloja koristi već izračunate parcijalne derivacije sloja iznad i tako smanjuje vrijeme potrebno za izračun.

Idući korak je promjena parametara mreže. Svaki se parametar mijenja u ovisnosti o stopi učenja η koja kontrolira jačinu promjene. Za ovaj izračun koristi se optimizator Adam koji prilagođava stopu učenja za svaki parametar posebno, umjesto da ju drži konstantnom. Adam je korišten jer pokazuje bolje rezultate u odnosu na druge metode, poput brže konvergencije u minimum.

Kako bi učenje bilo vremenski efikasno, koristi se paralelizacija. Međutim, za velike skupove podataka to nije moguće. Rješenje su mini grupe (engl. *mini batches*): obrada više podataka paralelno (u grupi) i sukladno njima promijeniti parametre mreže. Prolaskom kroz sve mini grupe završava jedna epoha učenja.

5. Implementacija

Za ovaj rad bila su učena tri modela. Sva tri modela imaju arhitekturu konvolucijskih mreža.

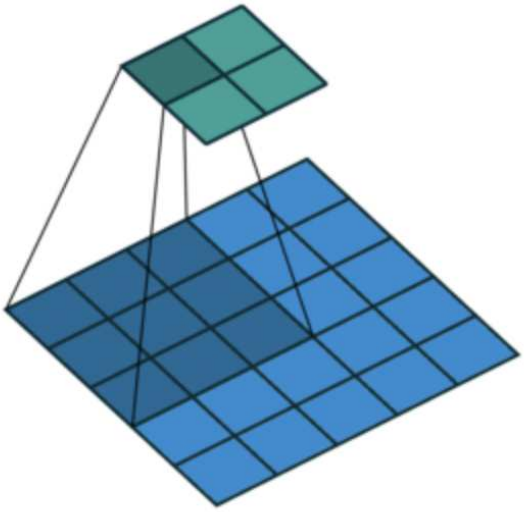
5.1. Konvolucijske mreže

Klasične neuronske mreže kojima su neuroni susjednih slojeva potpuno povezani nisu praktične zbog količine veza što vodi do kompleksnog i dugog učenja. Pogotovo kad su ulazni podaci slike, koje same po sebi imaju velike dimenzije (npr. slike u boji, slike visoke rezolucije). Također, kod obrade slika prostorna udaljenost ulaza ne može se zanemariti kao kod npr. modela koji kao ulaz prima dvije nezavisne karakteristike. Stvar je u tome da su bliži pikseli (ulazi u mrežu) korelirani jer vrlo vjerojatno pripadaju istom objektu na slici. Kod klasičnih neuronskih mreža ne postoji način da mreža koristi tu informaciju. Uz to, slike su otporne na transformacije poput translacije ulijevo, a klasične neuronske mreže u tom slučaju moraju nanovo učiti sve parametre [21].

Konvolucijske mreže su unaprijedne neuronske mreže gdje ne postoji veza između svih neurona u susjednim slojevima, već rade operaciju konvolucije nad skupinom neurona. Koriste tzv. filtar ili jezgru (engl. *kernel*) koja prolazi po neuronima i obavljajući operaciju konvolucije, stvara mapu značajki (engl. *feature map, channel*). Ovisno o vrsti konvolucije (1D, 2D, 3D) jezgra ima različite dimenzije. Za 2D konvoluciju koja se koristi nad slikama, jezgra je matrica koja sadrži težine. Operacija konvolucije zapravo je težinska suma korespondentnih izlaza neurona i težina, s aktivacijskom funkcijom (najčešće ReLU [22]), slika 5.1.

Nakon konvolucijskih slojeva često slijede slojevi poput:

1. Normalizacijski sloj – omogućava bržu konvergenciju; učenje je često stabilnije jer



Slika 5.1. "Animation of a variation of the convolution operation" by Vincent Dumoulin and Francesco Visin, licensed under MIT. Available at: `https://commons.wikimedia.org/wiki/File:Convolution_arithmetic_-_No_padding_strides.gif`

su težine svakog sloja učene uz manju ovisnost o drugim slojevima.

2. Slojevi sažimanja (engl. *pooling layers*) – smanjuju dimenzionalnost, postižu djelomičnu invarijantnost modela na translaciju. Obavljaju sažimanje uz pomoć jezgre koja određuje nove dimenzije sloja, npr.

1. Sažimanje najvećom vrijednošću (engl. *max pooling*) – od svih neurona bira onaj s najvećom izlaznom vrijednosti

2. Sažimanje srednjom vrijednošću (engl. *average pooling*) – uzima srednju vrijednost neurona

Postoje i slojevi isključivanja neurona (engl. *dropout*) koji nasumičnim odabirom postavljaju izlazne vrijednosti neurona na 0 kao pokušaj sprečavanja prenaučenosti.

5.2. Fourierova transformacija

Fourierova transformacija je matematički alat (formula) koji transformira signal iz vremenske domene u frekvencijsku domenu. Vremenska domena prikazuje promjenu signala kroz vrijeme, dok frekvencijska domena prikazuje amplitudu pojedine frekvencije u signalu. Fourierova transformacija omogućava dublju analizu signala koja nije moguća

dok se promatra samo promjena signala kroz vrijeme.

Fourierova transformacija na vremenskom otvoru (engl. *short-time Fourier transform*) je transformacija signala u vidu kako se on mijenja u vremenu. Za razliku od Fourierove transformacije, koja prikazuje amplitude frekvencija kroz cijeli signal, STFT dijeli signal na više dijelova (engl. *frames*), nad kojima se onda zasebno vrši Fourierova transformacija.

5.3. Modeli

Svaki od tri modela ima isti životni ciklus. Prvo je potrebno dobiti labele za sva tri skupa podataka (učenje, provjera, test). Klasa "engleski jezik" ima labelu 1, "španjolski jezik" labelu 2, a "njemački jezik" labelu 3. Nakon toga slijedi učitavanje podataka, koje je detaljnije objašnjeno u poglavljima za pojedini model: 5.4., 5.5., 5.6. Svaki audio zapis učitan je s frekvencijom uzorkovanja (engl. *sample rate*) 16 kHz te je u slučaju da je njegova duljina manja od 10 sekundi, dodana dopuna s nulama do 10 sekundi, a u slučaju da mu je duljina veća od 10 sekundi, odrezan je do točno 10 sekundi. Budući da je skup podataka velik i svaki zvučni zapis traje 10 sekundi, nije moguće učitati čitav skup podataka u memoriju odjednom. Zbog toga se koristi generator (prilagođen od koda [23]), koji omogućava učitavanje mini grupe u trenutku kad je potrebno. Slijedi učenje modela te njegova evaluacija. Kao mjeru kvalitete modela korištena mjera točnosti i utežana mjera F1, a kao prikaz korištena je matrica konfuzije. Da bi se moglo izračunati ove mjere, potrebno je izračunati četiri moguća ishoda klasifikacije za svaku pojedinu klasu:

True positive (TP) – broj primjera točno klasificiranih kao pripadnik toj klasi

True negative (TN) – broj primjera točno klasificiranih kao ne-pripadnik toj klasi

False positive (FP) – broj primjera netočno klasificiranih kao pripadnik toj klasi

False negative (FN) – broj primjera netočno klasificiranih kao ne-pripadnik toj klasi

Točnost se izražava kao

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} TP_i}{\text{Total Predictions}} \times 100\% \tag{5.1}$$

gdje je N broj klasa, a *Total predictions* ukupan broj testiranih primjera. Točnost je bila smatrana dovoljnom mjerom za procjenu kvalitete modela, ali shvatilo se da to nije uvijek slučaj. Primjer je neuravnotežen skup podataka, koji ima npr. 85% primjeraka jedne klase, a ukupno 15% primjeraka iz druge dvije klase. Model će naučiti da će postići jako dobru točnost (85%) ako samo predvidi da su svi primjerci primjerci prve klase. Točnost ovog modela je visoka, ali očigledno je da on nije od neke koristi. Zato se uz to može koristiti mjera F1.

Mjera F1 uravnotežuje mjeru točnosti jer umjesto da gleda ukupno po klasama, svaka klasa pridonosi. Mjera F1 svake klase izražava se ovako:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.2}$$

gdje su Precision (preciznost) i Recall (odziv) također mjere:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.4}$$

U ovom radu korištena je utežana mjera F1 (engl. *weighted F1 score*) koja se izražava ovako

$$\text{Weighted F1} = \sum_{i=1}^{N} \left(\frac{n_i}{N} \cdot F1_i \right) \tag{5.5}$$

gdje je n_i broj primjera klase i, a $F1_i$ F1 mjera klase i

Matrica konfuzije je matrica koja prikazuje sažeti zapis ponašanja modela, svaki red i u matrici čine primjeri koji su primjerci klase i raspoređeni po stupcima j po tome u koju ih je klasu model svrstao.

5.4. CNN i spektrogram

Kao prvi model, učena je konvolucijska neuronska mreža koja kao ulaze prima spektrograme zvučnih zapisa. Na svaki učitani zapis primjenjuje se STFT s veličinom okvira 255 i korakom okvira 128 kako bismo dobili spektrogram. Dalje se uzimaju apsolutne vrijednosti jer faza generalno nije bitna te se dodaje dimenzija kako bi dimenzija ulaza bile u skladu s onima koje CNN traži. Dalje, spektrogramu je promijenjena veličina na željene dimenzije kako bi učenje i potrošnja memorije bili efikasnije. Na kraju, spektrogram je normaliziran.

Konkretna CNN se sastoji od ulaznog sloja, normalizacijskog sloja, konvolucijskog sloja s 32 jezgre veličine 3x3 i aktivacijskom funkcijom ReLU, konvolucijskog sloja sa 64 jezgre veličine 3x3 i aktivacijskom funkcijom ReLU, sloja maksimalnog sažimanja s jezgrom veličine 2x2, konvolucijskog sloja sa 128 jezgri veličine 3x3 i aktivacijskom funkcijom ReLU, sloja maksimalnog sažimanja s jezgrom veličine 2x2, sloja isključivanja neurona koji isključuje 25% vrijednosti, sloja izravnavanja (engl. *flatten*) u 1D vektor, potpuno povezanog sloja sa 128 neurona i aktivacijskom funkcijom ReLU, sloja isključivanja neurona koji isključuje 50% vrijednosti i izlaznog sloja s tri neurona koji odgovaraju trima klasama jezika, slika 5.2.

```
model = models.Sequential([
    layers.Input(shape=input_shape),
    layers.Resizing(32, 32),
    Normalization(),
    layers.Conv2D(32, 3, activation='relu'),
    layers.Conv2D(64, 3, activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(128, 3, activation='relu'),
    layers.MaxPooling2D(),
    layers.Dropout(0.25),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(3)  # 3 classes
])
```

Slika 5.2. Arhitektura klasične CNN, prilagođena verzija koda [24]

5.5. ResNet-50 i spektrogram

Kao drugi model, iskorištena je već prednaučena rezidualna konvolucijska neuronska mreža ResNet-50 koja je onda precizno podešena (engl. *fine tuned*) na ovom skupu podataka. To je mreža s 50 slojeva: 48 konvolucijskih, jedan sloj maksimalnog sažimanja i jedan sloj prosječnog sažimanja. Pokazala je jako dobre rezultate u području klasifikacije slika pa se ovdje koristi sa spektrogramima. Jedina promjena je što su spektrogrami jednokanalni (engl. *grayscale*), a ulaz u ResNet-50 su slike u boji, odnosno one s tri kanala. Učitavanje podataka je isto kao i kod prvog modela, samo se prije normalizacije spektrogram pretvara u sliku u boji.

Precizno podešavanje modela sastoji se od sloja izravnavanja u 1D vektor, potpuno povezanog sloja s 512 neurona i aktivacijskom funkcijom ReLU, normalizacijskog sloja, sloja isključivanja neurona koji isključuje 50% vrijednosti, potpuno povezanog sloja s 256 neurona i aktivacijskom funkcijom ReLU, normalizacijskog sloja, sloja isključivanja neurona koji isključuje 50% vrijednosti i izlaznog sloja s tri neurona, slika 5.3.

```
imported_model = tf.keras.applications.ResNet50(include_top=False, input_shape=input_shape, pooling='avg', weights='imagenet')

for layer in imported_model.layers:
    layer.trainable = False

model = Sequential()
model.add(imported_model)
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.5))
model.add(layers.Dense(256, activation='relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.5))
model.add(layers.Dense(3, activation='softmax'))  # 3 classes
```

Slika 5.3. Arhitektura precizno podešene mreže ResNet-50, prilagođena verzija koda [25]

5.6. CNN i značajke MFCC

Još jedan pristup problemu sa zvučnim zapisima je korištenje značajki MFCC.

5.6.1. MFCC

Kepstralni koeficijenti mel-frekvencije (MFCC) su reprezentacija zvučnog signala na način sličan ljudskom uhu. Ljudsko uho razliku u frekvencijama ne percipira linearno; razlike između nižih frekvencija su puno istaknutije od viših. U istraživanjima se najčešće

koristi prvih 13 vrijednosti jer predstavlja dobar omjer količine informacije i kompleksnosti računanja.

Za svaki je zapis izlučeno prvih 13 značajki MFCC koje su onda normalizirane. Sama konvolucijska mreža sastoji se od konvolucijskog sloja s 32 jezgre veličine 2x2 i aktivacijskom funkcijom ReLU, sloja maksimalnog sažimanja s jezgrom veličine 2x2, konvolucijskog sloja s 32 jezgre veličine 2x2 i aktivacijskom funkcijom ReLU, sloja maksimalnog sažimanja s jezgrom veličine 2x2, sloja izravnavanja u 1D vektor, potpuno povezanog sloja sa 64 neurona i aktivacijskom funkcijom ReLU, sloja isključivanja neurona koji isključuje 50% vrijednosti i izlaznog sloja s tri neurona.

```
model = models.Sequential()
model.add(layers.Conv2D(32, (2, 2), activation='relu', input_shape=sample_shape))
model.add(layers.MaxPooling2D(pool_size=(2, 2)))
model.add(layers.Conv2D(32, (2, 2), activation='relu'))
model.add(layers.MaxPooling2D(pool_size=(2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(3, activation='softmax'))
```

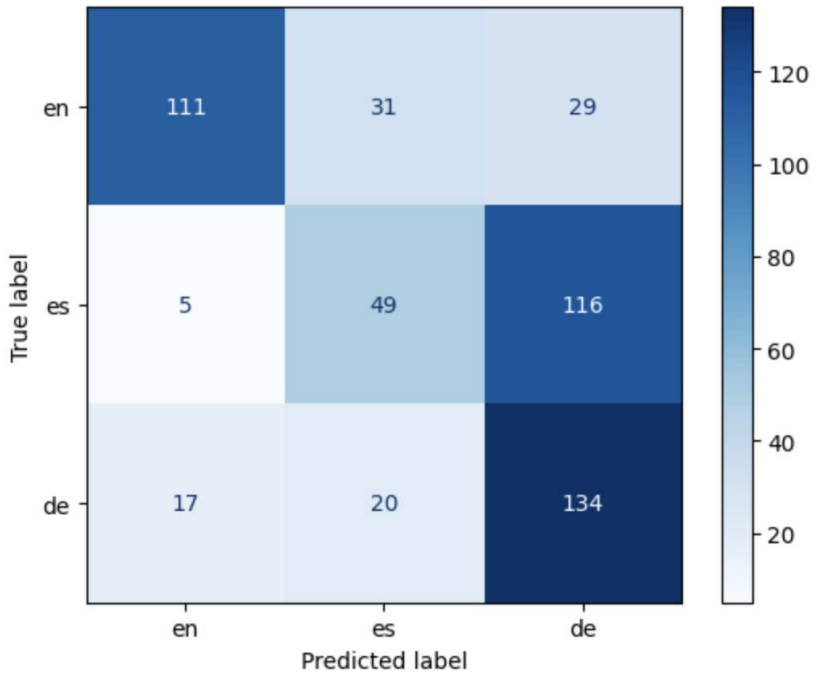
Slika 5.4. Arhitektura mreže sa značajkama MFCC, prilagođena verzija koda [26]

6. Rezultati i rasprava

U tablici 6.1. te na slikama 6.1. – 6.3. dani su rezultati klasifikacije modela na skupu za testiranje.

Model	Točnost	Utežana mjera F1
MFCC model	83%	83%
Klasična CNN	57%	56%
ResNet-50	52%	54%

Tablica 6.1. Rezultati modela na skupu za testiranje

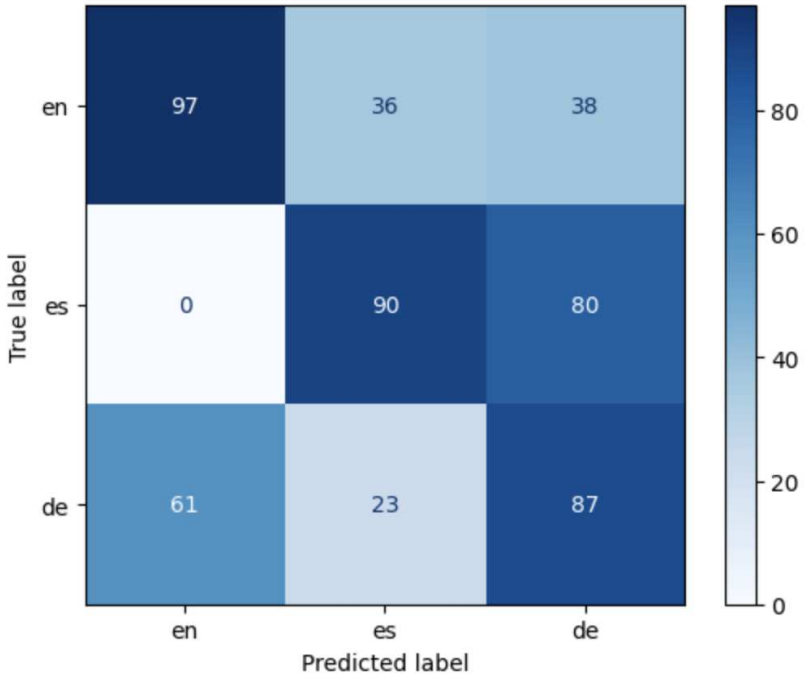


Slika 6.1. Matrica konfuzije klasične CNN

Matrica konfuzije prvog modela 6.1. i tablica preciznosti prvog modela 6.2. pokazuju kako model najpreciznije predviđa engleski jezik, dok je preciznost najlošija za njemački. Od 279 puta gdje je predviđen njemački, čak 116 puta (41%) je bila riječ o španjolskom.

Jezik	Preciznost
engleski	84%
španjolski	49%
njemački	48%

Tablica 6.2. Preciznost po klasama za model klasične CNN

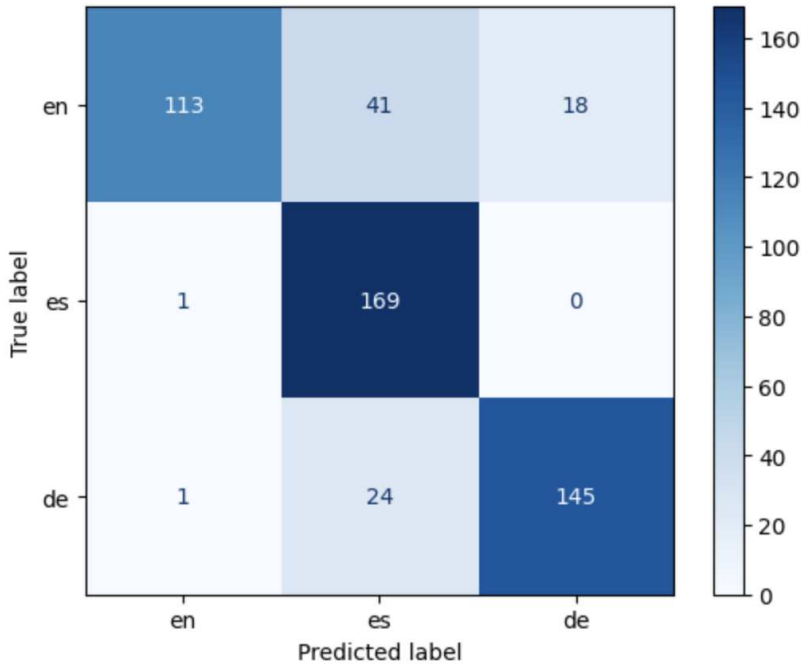


Slika 6.2. Matrica konfuzije ResNet-50 modela

Jezik	Preciznost
engleski	61%
španjolski	60%
njemački	42%

Tablica 6.3. Preciznost po klasama za model ResNet-50

Matrica konfuzije drugog modela 6.2. i tablica preciznosti drugog modela 6.3. pokazuju kako model podjednako precizno predviđa engleski i španjolski jezik, dok je njemački opet problematičan. U ovom modelu od 205 puta gdje je predviđen njemački, 80 puta (39%) je bila riječ o španjolskom. Ovaj model, kao i prvi ima nisku točnost i utežanu mjeru F1.



Slika 6.3. Matrica konfuzije modela sa značajkama MFCC

Jezik	Preciznost
engleski	98%
španjolski	72%
njemački	89%

Tablica 6.4. Preciznost po klasama za model sa značajkama MFCC

Matrica konfuzije trećeg modela 6.3. i tablica preciznosti trećeg modela 6.4. pokazuju kako model ponovno najpreciznije predviđa engleski jezik, a ono što je novo je preciznost njemačkog jezika koja je značajno porasla. Ovaj model je bolji od prva dva u svakoj mjeri.

Vidimo da prva dva modela nisu pokazala dobre rezultate, što upućuje na to da mreže nisu dovoljne kompleksnosti ili da se u spektrogramu ne mogu dovoljno dobro prikazati sve značajke zvuka koje bi bile potrebne za ovakav zadatak. S druge strane, treći model koji koristi značajke MFCC kao ulaze postiže vrlo dobre rezultate: točnost i utežanu mjeru F1 od 83%. To nije začuđujuće jer su značajke MFCC specifično konstruirane da uhvate značajke ljudskog govora, dok su spektrogrami detaljni i mogu zahvatiti pozadinsku buku i značajke nebitne za zadatke vezane uz ljudski govor. Značajke MFCC također zato omogućavaju i bržu konvergenciju, što je i slučaj u ovom radu gdje su prva dva modela učena u deset epoha, a treći u pet.

7. Zaključak

Cilj ovog završnog rada bio je razviti model umjetne inteligencije za što točnije raspoznavanje jezika. Modeli bi bili korisni na mjestima gdje je potrebna automatizirana komunikacija s ljudima, poput prilagodljivih grafičkih sučelja u aplikacijama za bolje iskustvo ljudi.

U ovom istraživanju ispitana su tri modela dubokog učenja, svaki zasnovan na konvolucijskoj mreži. Prvi i drugi su radili sa spektrogramima, ali su arhitekture mreža različite. Prvi je gradio konvolucijsku mrežu od početka dok je drugi koristio već unaprijed pripremljenu konvolucijsku mrežu ResNet-50 koju je onda fino podešavao. Treći je radio sa značajkama MFCC. Od ta tri, jedino je treći imao zadovoljavajuće rezultate u aspektu točnosti i utežane mjere F1 od 83%.

Mogućnosti poboljšanja rezultata postoje, ali su mnoge računalno zahtjevne. Može se povećati kompleksnost mreže dodatkom slojeva što u pravilu poboljšava rezultate, ali postoji rizik od prenaučenosti ako je zadatak nedovoljno kompleksan u odnosu na mrežu. Uz to, mogu se istražiti druge značajke za ulaz u mrežu, poput kromatskih značajki audio zapisa (engl. *chroma features*) [27].

Literatura

[1] W. contributors, “Natural language processing”, https://en.wikipedia.org/wiki/Natural_language_processing, June 2024., pristupljeno: 1.6.2024.

[2] GeeksforGeeks, “Impact of dataset size on deep learning model”, https://www.geeksforgeeks.org/impact-of-dataset-size-on-deep-learning-model/, April 2024., pristupljeno: 3.6.2024.

[3] Tableau Software, “Natural language processing examples”, https://www.tableau.com/learn/articles/natural-language-processing-examples, June 2024., pristupljeno: 1.6.2024.

[4] A. Ali, “Understanding the nlp pipeline: A comprehensive guide”, https://medium.com/@asjad_ali/understanding-the-nlp-pipeline-a-comprehensive-guide-828b2b3cd4e2, January 2024., pristupljeno: 1.6.2024.

[5] A. Velankar, H. Patil, i R. Joshi, “Mono vs multilingual bert for hate speech detection and text classification: A case study in marathi”, *https://arxiv.org/pdf/2204.08669*, 2022.

[6] S. Agrawal, “Characteristics of ‘bad data’ for machine learning and potential solutions”, https://medium.com/@sanidhyaagrawal08/characteristics-of-bad-data-for-machine-learning-and-potential-solutions-88760bfa1532, December 2021., pristupljeno: 2.6.2024.

[7] M. C. Voice, “Mozilla common voice”, https://commonvoice.mozilla.org/en/datasets, n.d., pristupljeno: 2.6.2024.

[8] Kaggle, “Your machine learning and data science community”, https://www.kaggle.com/, n.d., pristupljeno: 2.6.2024.

[9] ——, “Spoken language identification”, https://www.kaggle.com/datasets/toponowicz/spoken-language-identification/data, July 2018., pristupljeno: 2.6.2024.

[10] TopCoder, “Topcoder challenge listings”, https://www.topcoder.com/challenges?tracks[DS]=true&tracks[Des]=true&tracks[Dev]=true&tracks[QA]=true&types[]=CH&types[]=F2F&types[]=MM&types[]=TSK, n.d., pristupljeno: 2.6.2024.

[11] Kaggle, “Spoken languages”, https://www.kaggle.com/datasets/mittalshubham/spoken-languages, April 2019., pristupljeno: 2.6.2024.

[12] Ar5iv, “How to avoid machine learning pitfalls: a guide for academic researchers”, https://ar5iv.labs.arxiv.org/html/2108.02497#S2.SS3, n.d., pristupljeno: 2.6.2024.

[13] W. contributors, “Machine learning”, https://en.wikipedia.org/wiki/Machine_learning, June 2024., pristupljeno: 3.6.2024.

[14] M. A. H. C. J. P. I. H. Witten, E. Frank, *Data mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2016.

[15] B. Dalbelo Bašić i J. Šnajder, “Strojno učenje”, https://www.fer.unizg.hr/_download/repository/UI-2020-10-StrojnoUcenje.pdf, 2020., pristupljeno: 3.6.2024.

[16] W. contributors, “Neuron”, https://en.wikipedia.org/wiki/Neuron, May 2024., pristupljeno: 4.6.2024.

[17] ——, “Neurotransmitter”, https://en.wikipedia.org/wiki/Neurotransmitter, June 2024., pristupljeno: 4.6.2024.

[18] ——, “Deep learning”, https://en.wikipedia.org/wiki/Deep_learning, June 2024., pristupljeno: 5.6.2024.

[19] “How does deep learning enable more accurate predictions?” https://www.linkedin.com/advice/3/how-does-deep-learning-enable-more-accurate-predictions-ia9tf, December 2023., pristupljeno: 5.6.2024.

[20] A. Acharya, “Training, validation, test split for machine learning datasets”, https://encord.com/blog/train-val-test-split/, April 2024., pristupljeno: 2024-06-11.

[21] S. J. Prince, *Understanding Deep learning*. MIT Press, 2023.

[22] W. contributors, “Convolutional neural network”, https://en.wikipedia.org/wiki/Convolutional_neural_network, June 2024., pristupljeno: 6.6.2024.

[23] “A detailed example of data generators with keras”, https://stanford.edu/~shervine/blog/keras-how-to-generate-data-on-the-fly, n.d., pristupljeno: 10.5.2024.

[24] “Simple audio recognition: Recognizing keywords”, https://www.tensorflow.org/tutorials/audio/simple_audio, n.d., pristupljeno: 10.5.2024.

[25] B. Wasike, “How to build a deep learning model with keras and resnet-50”, https://medium.com/@bravinwasike18/building-a-deep-learning-model-with-keras-and-resnet-50-9dd6f4eb3351, May 2023., pristupljeno: 13.5.2024.

[26] “Tensorflow lite tutorial part 2: Speech recognition model training”, https://www.digikey.com/en/maker/projects/tensorflow-lite-tutorial-part-2-speech-recognition-model-training/d8d04a2b60a442cf8c3fa5c0dd2a292b, n.d., pristupljeno: 20.5.2024.

[27] W. contributors, “Chroma feature”, https://en.wikipedia.org/wiki/Chroma_feature, February 2024., pristupljeno: 11.6.2024.

Sažetak

Raspoznavanje izgovorenog jezika iz kratkog zvučnog zapisa metodama strojnog učenja

Lana Bartolović

Ljudski jezik nije razumljiv računalu sam po sebi. Stoga se obradom prirodnog jezika nastoji premostiti ta prepreka. Jedan od temeljnih zadataka je određivanje jezika. U ovom radu su učena i ispitana tri modela dubokog učenja s ciljem određivanja uspješnosti klasificiranja jezika iz zvučnog zapisa. Najbolji model, zasnovan na konvolucijskoj mreži i značajkama MFCC, postigao je mjeru točnosti od 83% i utežanu mjeru F1 od 83%

Ključne riječi: obrada prirodnog jezika, duboko učenje, Fourierova transformacija, spektrogram, MFCC, konvolucijska neuronska mreža, klasifikacija

Abstract

Recognition of spoken language from a short audio recording using machine learning methods

Lana Bartolović

The human language is not understandable to computers in itself. Therefore, natural language processing aims to bridge this gap. One of the fundamental tasks is language identification from audio recordings. In this work, three deep learning models were trained and tested with the goal to determine their success in classifying languages from audio recordings. The best model, based on convolutional network and MFCC features, achieved an accuracy rate of 83% and a weighted F1 score of 83%.

Keywords: natural language processing, deep learning, Fourier transformation, spectrogram, MFCC, convolutional neural network, classification