

Klasifikacija zvuka primjenom prijenosa znanja

Baričević, Mateo

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:996277>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-03**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



Sveučilište u Rijeci – Fakultet informatike i digitalnih tehnologija

Diplomski sveučilišni studij informatike
Informacijski i komunikacijski sustavi

Mateo Baričević

Klasifikacija zvuka primjenom prijenosa znanja

Diplomski rad

Mentor: prof. dr. sc. Maja Matetić

Rijeka, prosinac 2022.

Rijeka, 27. travanj 2022.

Zadatak za diplomski rad

Pristupnik: **Mateo Baričević**

Naziv diplomskog rada: **Klasifikacija zvuka primjenom prijenosa znanja**

Naziv diplomskog rada na eng. jeziku: **Audio classification using transfer learning**

Sadržaj zadatka:

Zadatak diplomskog rada je primijeniti metodu prijenosa znanja u zadatku klasifikacije emocija audio zapisa ljudskog govora. Potrebno je istražiti da li model temeljen na prijenosu znanja daje bolje rezultate s obzirom na to da posjeduje znanje o raspoznavanju govora. Potrebno je usporediti kvalitetu modela za detekciju emocija tradicionalnim načinom učenja i modela dobivenog predtreniranjem na skupu podataka za raspoznavanje riječi i naknadno dotreniranog na skupu podataka za emocije. U radu treba dati pregled primjera primjene postupka prijenosa znanja, opisati u radu korištenu metodologiju, analizu podataka i rezultate.

Mentor:
Prof. dr. sc. Maja Matetić



Voditeljica za diplomske radove:
Prof. dr. sc. Ana Meštrović



Komentor:

Zadatak preuzet: 27. travnja 2022.



(potpis pristupnika)

Sažetak

Prikupljanje skupova označenih podataka je u praksi vrlo zahtjevno ili ponekad i nemoguće, a da u isto vrijeme sadrže dovoljno podataka za uspješno učenje modela strojnog učenja. Jedno od mogućih rješenja ovog problema je učenje prijenosom znanja. U ovome radu je istraženo koliko je ova tehnika učenja uspješna za primjenu u domeni klasifikacije zvuka i pod kojim uvjetima. Učinci učenja prijenosom znanja promatrani su u smislu prediktivne izvedbe modela koji su evaluirani F_1 , MCC i ROC-AUC metrikama vrednovanja. U eksperimentu je korišten skup podataka koji ima manji broj označenih instanci kako bi se usporedilo tradicionalno učenje nad takvim skupom podataka od onog gdje se koristila tehnika učenja prijenosom znanja. Koristeći skup podataka koji ima veći broj označenih instanci, naučen je izvorni model iz kojeg je preneseno znanje u obliku težina na neuronima. Eksperiment se sastojao od pet različitih scenarija, a u svakom od njih je prenesena različita količina znanja u obliku težina iz određenih slojeva. Prijenos znanja se odvijao unutar iste domene primjene (klasifikacija zvuka). Promatrana je prediktivna izvedba svih modela. Kako bi se potvrdila opažanja u dobivenim rezultatima, primijenjen je statistički test kako bismo potvrdili značajnost rezultata. Opći zaključak ovog istraživanja je da učenje prijenosom znanja unutar domene klasifikacije zvuka može pozitivno utjecati na prediktivnu izvedbu modela.

Ključne riječi: strojno učenje, učenje prijenosom znanja, klasifikacija zvuka

Sadržaj

Sažetak	I
Sadržaj.....	II
1. Uvod.....	1
2. Srodna literatura	2
3. Metodologija	4
3.1. Strojno učenje i neuronske mreže	4
3.2. Učenje.....	7
3.3. Metrike vrednovanja	8
3.3.1. Točnost	8
3.3.2. <i>F</i> -rezultat	8
3.3.3. Matthewsov koeficijent korelacije	9
3.3.4. Površina ispod krivulje radne karakteristike prijamnika.....	9
3.4. Učenje prijenosom znanja	10
3.5. Unakrsna provjera valjanosti.....	11
4. Skupovi podataka	14
4.1. Speech Commands	14
4.2. RAVDESS.....	14
4.3. Usporedba skupova podataka	14
4.4. Standardizacija podataka.....	15
5. Analiza podataka i rezultati.....	16
5.1. Postavke eksperimenta	16
5.1.1. Aktivacijska funkcija.....	16
5.1.2. Funkcija gubitaka	17
5.1.3. Optimizator.....	18
5.1.4. Unakrsna provjera valjanosti.....	18
5.2. Proces učenja prijenosom znanja	18
5.3. Rezultati	20
6. Zaključak.....	22
Popis slika	24
Popis tablica	24
Popis priloga.....	24

1. Uvod

Iako je tradicionalno strojno učenje postiglo veliki uspjeh i uspješno se primjenjuje u mnogim praktičnim primjenama, još uvijek ima nekih ograničenja u određenim situacijama u stvarnom svijetu. U idealnom svijetu, skup podataka za strojno učenje bi sadržavao veliku količinu označenih instanci koje imaju istu distribuciju kao podaci za testiranje. Međutim, prikupljanje dovoljno podataka za učenje je često skupo, dugotrajno ili čak nerealno u mnogim situacijama.

Polu-nadzirano učenje može djelomično riješiti ovaj problem smanjivanjem potrebe za velikom količinom podataka jer takav pristup zahtijeva samo ograničen broj označenih podataka i koristi veliku količinu neoznačenih podataka za poboljšanje točnosti rezultirajućeg modela. Ovim putem se umanjuje problem kada u skupu podataka nedostaje označenih instanci, no i dalje postoji mogućnost prisutnosti problema gdje je teško prikupiti čak i neoznačene podatke. Druga metoda za rješavanje ovog problema je kroz povećanje podataka (engl. *data augmentation*). Ako skup podataka sadrži slike, najpopularnija trenutna praksa za povećanje podataka je izvođenje kombinacije afinih transformacija slike i modifikacije boja. Neke od najkorištenijih afinih transformacija su rotacija, refleksija, skaliranje i smicanje. Ovaj pristup se može koristiti samo u aplikacijama računalnog vida (engl. *computer vision*) i nije primjenjiv u ostalim situacijama gdje se skup podataka ne tvori od slika.

Tema učenja prijenosom znanja je relativno dobro istraženo područje kroz jako puno akademskih radova koji se bave ovom temom u posljednjem desetljeću. Cilj ovakvog učenja je poboljšati učenje u ciljnom zadatku tako da se koristi znanje koje je prethodno naučeno iz izvornog zadatka gdje je rezultat tog učenja bio dobar zbog pristupa većoj količini podataka u toj domeni. Znanje koje se prenosi sadržano je u težinama koje je potrebno prenijeti u novi model te ih zaključati kako se ne bi mijenjale dok se model uči na skupu za učenje. Nakon učenja se može izvršiti ugađanje težina na određinom zadatku (engl. *fine-tuning*) gdje se otključaju svi slojevi i time se model dodatno ugađa podacima iz ciljne domene.

Učenje prijenosom znanja primijenjeno je na mnoge aplikacije u stvarnom svijetu. Postoji niz primjera primjene koji se odnose na obradu prirodnog jezika, točnije u područjima klasifikacije emocija, klasifikacije teksta, otkrivanja neželjene e-pošte i klasifikacije teksta na više jezika. Ostale aplikacije učenja prijenosom znanja uključuju klasifikaciju slika i klasifikaciju video koncepta. Postoji nekoliko specifičnih primjena učenja prijenosom znanja koje su obrađene u drugim radovima, no općenito su rješenja generička, što znači da se rješenje može lako primijeniti na slučajeve za koje ne postoje implementacije i koje nisu testirane u drugim radovima, te se najviše koristi u području obrade prirodnog jezika i obrade slike. Iako je konceptualno jednostavno primijeniti učenje prijenosom znanja, u praksi se mogu pojaviti izazovi koji nisu trivijalni za riješiti.

U ovome radu fokusirao sam se na učenje prijenosom znanja u zadatku klasifikacije zvuka gdje su izvorna i ciljna domena iste. Zadatak izvornog modela je klasifikacija zvučnih zapisa izgovora 35 riječi engleskog jezika, a ciljnog modela je klasifikacija zvučnih zapisa izgovora dvije rečenice engleskog jezika u osam različitih emocija. Ovim eksperimentom htio bih ispitati primjenjivost tehnike učenja prijenosom znanja u zadatku klasifikacije zvuka.

2. Srodna literatura

Tehnike strojnog učenja i rudarenja podataka korištene su u brojnim aplikacijama u stvarnom svijetu. Pretpostavka tradicionalnih metodologija strojnog učenja jest da su podaci za učenje i podaci za testiranje preuzeti iz iste domene, tako da su prostor ulaznih značajki i karakteristike distribucije podataka jednaki. Međutim, u nekim scenarijima strojnog učenja u stvarnom svijetu ova pretpostavka ne vrijedi. Postoje slučajevi u kojima su podaci za učenje skupi i gdje ih je teško prikupiti. Zbog toga postoji potreba za stvaranjem modela visokih performansi učenjem s lakše dobivenim podacima iz različitih domena. Takva metodologija se naziva učenje prijenosom znanja. U literaturi su prikazane tehnike ovakve vrste učenja te je zaključeno da su rješenja za učenje prijenosom znanja neovisna o veličini podataka te da se mogu primijeniti na skupove velikih podataka (engl. *big data*) [1].

Učenjem neuronske mreže u kojem prenosimo znanje iz već naučene mreže za srodni zadatak dolazi do poboljšanja performansi mreže u novom zadatku gdje možda nemamo dovoljno podataka da mreža nakon učenja daje zadovoljavajuću prediktivnu točnost. Iako je većina algoritama strojnog učenja dizajnirana za rješavanje pojedinačnih zadataka, razvoj algoritama koji olakšavaju prijenos učenja je tema koja neprestano zanima zajednicu strojnog učenja. U literaturi je prikazan uvod u ciljeve, formulacije i izazove učenja prijenosom znanja te pregled trenutnog stanja tehnika ovakvog učenja. Također je prikazan prijenos znanja u induktivnom učenju (engl. *inductive learning*) i ojačanom učenju te je detaljno razrađeno pitanje mapiranja zadataka [2].

Zbog širokih mogućnosti primjene, učenje prijenosom znanja postalo je popularno i obećavajuće područje u strojnom učenju. Zbog brzog širenja područja učenja prijenosom znanja, u literaturi je prikazana povezanost i sistematiziranost postojećih istraživanja ovakvog učenja, kao i sažet prikaz i interpretacija mehanizma i strategije učenja prijenosa znanja na sveobuhvatan način. Također je prikazan pregled velikog broja reprezentativnih pristupa ovakvoj vrsti učenja, posebno homogenih pristupa učenju prijenosom znanja, iz perspektive podataka i modela. Eksperimentalni rezultati pokazuju važnost odabira odgovarajućih modela učenja prijenosa znanja za različite primjene u praksi [3].

Glavna pretpostavka u mnogim algoritmima strojnog učenja i rudarenja podataka je da podaci za učenje i budući podaci moraju biti u istom prostoru značajki i da moraju imati istu distribuciju. Međutim, u mnogim stvarnim aplikacijama ova pretpostavka možda neće vrijediti. Na primjer, kada se radi o zadatku klasifikacije u jednoj domeni od interesa, ali imamo dovoljno podataka za učenje samo u drugoj domeni od interesa, gdje budući podaci mogu biti u drugom prostoru značajki ili slijede drugačiju distribuciju podataka. U takvim bi slučajevima učenje prijenosom znanja, ako se uspješno izvede, uvelike poboljšalo izvedbu učenja izbjegavanjem skupog napora dobivanja podataka u inicijalnoj domeni od interesa. Posljednjih se godina učenje prijenosom znanja sve češće počelo koristiti za rješavanje ovog problema. Literatura se usredotočuje na kategorizaciju i pregled trenutnog napretka u ovoj tehnici učenja kod zadataka klasifikacije, regresije i grupiranja. Također, raspravlja o odnosu između učenja prijenosa znanja i drugih srodnih tehnika strojnog učenja kao što su prilagodba domene (engl. *domain adaptation*), višezadačno učenje (engl. *multitask learning*) i pristranost odabira uzorka (engl. *sample selection bias*), kao i pomak kovarijata (engl. *covariate shift*) [4].

U literaturi se koristi ImageNet, prethodno naučeni standardni duboki model konvolucijske neuronske mreže koristeći tehniku učenja prijenosom znanja u zadatku audio klasifikacije. Audio zapisi su pretvoreni u audio spektrograme i zato su mogli koristiti ovaj model za prijenos znanja. Iako postoji značajna razlika između audio spektrograma i standardnih ImageNet uzoraka slike, pretpostavke o prijenosu znanja i dalje vrijede. Literatura pokazuje da je korištenje unaprijed ugođenih težina iz ImageNet modela bolje od korištenja nasumično

inicijaliziranih vrijednosti težina kako bi nova konvolucijska neuronska mreža bolje naučila klasificirati audio zapise iz njihovih spektrograma vizualizacijom gradijenata. Osim toga, pokazuje se da postoji varijacija u performansama u različitim izvedbama učenja istog modela iako se koriste unaprijed ugođene težine modela za inicijalizaciju novog modela. Ova varijacija nastaje zbog nasumičnog pokretanja sloja linearne klasifikacije i nasumične podjele mini-serija u višestrukim izvođenjima. To donosi značajnu raznolikost za izradu jačih skupnih modela s ukupnim poboljšanjem točnosti. Rezultati pokazuju da DenseNet koji koristi znanje iz ImageNet modela postiže točnost od 92.89% na ESC-50 skupu podataka i točnost od 87.42% na UrbanSound8K skupu podataka [5].

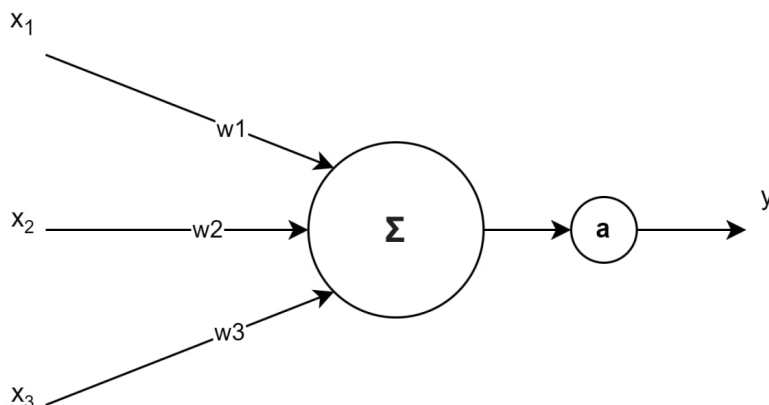
3. Metodologija

3.1. Strojno učenje i neuronske mreže

Strojno učenje je grana računarstva koja istražuje algoritme koji na temelju dostupnih podataka uče dati točan odgovor, tj. poboljšavaju svoju točnost za određeni zadatak. Ovo uključuje tradicionalne pristupe kao što su nasumična šuma (engl. *random forest*), linearna regresija, metoda potpornih vektora (engl. *support vector machine*), ali i modernije pristupe poput umjetnih neuronskih mreža. Primjene strojnog učenja se ovisno o vrsti problema mogu podijeliti u tri skupine: učenje pod nadzorom (engl. *supervised learning*), učenje bez nadzora (engl. *unsupervised learning*) i ojačano učenje (engl. *reinforcement learning*). Sve od ove tri metode kao svoj ulaz primaju vektor nezavisnih varijabli koje opisuju stanje na temelju kojeg se želi napraviti predviđanje, a te vrijednosti se često nazivaju značajkama ili atributima. Učenje pod nadzorom je prikladno za situacije kada su poznate nezavisne i zavisne varijable, a cilj je modelirati funkciju koja transformira nezavisne u zavisne varijable. Učenje bez nadzora je prikladno za situacije kada su poznate samo nezavisne varijable te je potrebno na temelju njih segmentirati podatke (engl. *clustering*). Ojačano učenje je primjenjivo u situacijama u kojima postoji interaktivno okruženje u kojem se mogu poduzimati akcije za koje se dobiva brožčana nagrada. U takvom okruženju, cilj algoritma strojnog učenja je pronaći strategiju, odnosno slijed akcija koji maksimizira dobivene nagrade.

Problemi učenja pod nadzorom se nadalje mogu podijeliti na probleme regresije i probleme klasifikacije. U regresijskim problemima zavisne varijable poprimaju kontinuirane vrijednosti što znači da i model mora biti prilagođen za predviđanje kontinuiranih vrijednosti. U klasifikacijskim problemima potrebno je zadani primjerak svrstati u pripadajuću klasu. Primjerice, prepoznavanje je li slika na sebi sadrži psa ili mačku je klasifikacijski problem jer slika ili sadrži ili ne sadrži traženi objekt. Ovaj primjer je podrazumijevao kako zadana slika može pripadati samo jednoj klasi, a to je i definicija binarne i višeklasne klasifikacije. Međutim, ponekada je potrebno moći zadani primjerak svrstati u više klasa istovremeno te se tada govori o klasifikaciji s više oznaka (engl. *multi-label classification*). U ovakvoj vrsti klasifikacije, zadatak je na temelju nezavisnih varijabli pridružiti zadanom primjerku oznake koje su istinite za njega. Na primjer, klasifikacija slike koja istovremeno sadrži i psa i mačku.

Modeliranje problema umjetnim neuronskim mrežama je tijekom posljednja dva desetljeća postalo sve popularnije zbog njihove ekspresivnosti i velikog napretka u pojedinim primjenama (npr. klasifikacija slika na *ImageNet* izazovu). Neuronske mreže se sastoje od neurona, a model jednog neurona s tri ulaza je prikazan na slici 1.



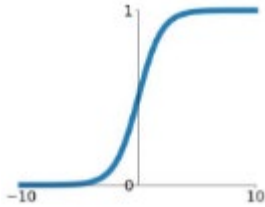
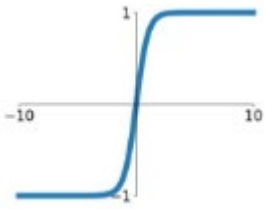
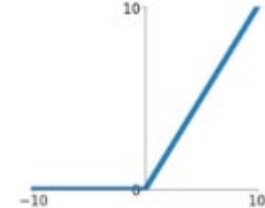
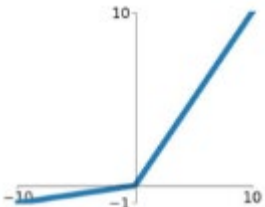
Slika 1. Model neurona s tri ulaza.

Ulazi u neuron su označeni s x , a izlaz iz neurona s y . Neuron funkcionira tako da se svaki ulaz x_i množi s pripadajućom težinom w_i . Tako dobivene vrijednosti se zbrajaju te se novo dobivena vrijednost koristi kao ulaz u aktivacijsku funkciju a . U konačnici, cijela operacija neurona se može matematički prikazati kao:

$$y = a(x_1 * w_1 + x_2 * w_2 + x_3 * w_3)$$

Aktivacijska funkcija ima veliku ulogu u sposobnosti neuronske mreže da modelira neku funkciju jer ona uvodi nelinearnosti te time omogućava neuronskoj mreži da modelira nelinearne funkcije. Postoje mnoge aktivacijske funkcije, a pregled najčešće korištenih je dan u tablici 1.

Tablica 1. Pregled najkorištenijih aktivacijskih funkcija [6].

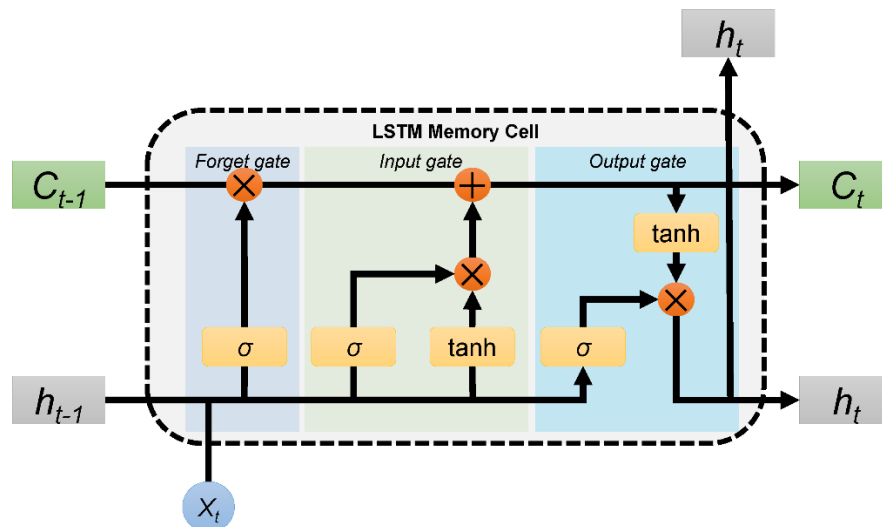
Naziv	Formula	Slika
Sigmoid	$a(x) = \frac{1}{1 + e^{-x}}$	
Tanh	$a(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	
ReLU	$a(x) = \max(0, x)$	
Leaky ReLU	$a(x) = \max(0.1x, x)$	

Neuronske mreže se obično grade od slojeva te su za potrebe ovog rada korištene četiri vrste slojeva: konvolucijski sloj, sloj sažimanja, povratni i potpuno povezani sloj. Konvolucijski slojevi se sastoje od konvolucijskih filtera koji u ulaznom signalu pronalaze određene uzorke. Pritom je svaki konvolucijski filter specijaliziran za točno jedan uzorak što znači da je potrebno odabrati dovoljno veliki broj filtera kako bi se iz ulaznog signala moglo prepoznati dovoljan broj uzoraka na temelju kojih se može napraviti predviđanje. Prilikom prelaženja konvolucijskih filtera preko ulaznog signala, na mjestima s većim poklapanjem između signala i traženog uzorka će doći do jače aktivacije, a na mjestima slabog ili nikakvog poklapanja do

slabe ili nikakve aktivacije. Može se reći da se konvolucijski slojevi koriste u svrhu „izvlačenja” značajki iz ulaznog signala (engl. *feature extraction*).

Vrlo često se nakon konvolucijskih slojeva postavlja sloj sažimanja kako bi se smanjila dimenzionalnost spajanjem susjednih vrijednosti u jednu (npr. u obradi slike to može biti spajanje više susjednih piksela u jednog). Sažimanje se uglavnom izvodi na temelju prosječne ili maksimalne vrijednosti, ali moguće su i druge varijacije. Za potrebe ovog rada korišteno je sažimanje na temelju maksimalne vrijednosti. To znači da će rezultirajuća vrijednost ove operacije odgovarati maksimalnoj vrijednosti ulaznih vrijednosti koje zahvaća trenutni prozor sažimanja.

Povratni slojevi poput LSTM i GRU slojeva omogućavaju obradu vremenskih serija pomoću neuronskih mreža te uvode vremensku dimenziju, a kao ulaz primaju vremenski niz kako su se značajke mijenjale kroz vrijeme. Za potrebe ovog diplomskog rada korišten je LSTM sloj te je shematski prikaz njega dan na slici 2. Izlaz kod ove vrste slojeva ne ovisi samo o trenutnom ulazu već i o prethodnim ulazima. U svakom vremenskom koraku t ovaj sloj izračunava dvije izlazne vrijednosti: izlaz h_t i stanje povratnog sloja c_t . Obje se vrijednosti zatim prebacuju na ulaz za idući vremenski korak te postaju h_{t+1} i c_{t+1} , a uz njih se dodaje i opservacija iz idućeg vremenskog koraka x_{t+1} .



Slika 2. Povratni LSTM stroj [7].

Potpuno povezani slojevi se sastoje od neurona, a nazivaju se potpuno povezani jer je svaki neuron iz tog sloja povezan sa svim izlaznim vrijednostima iz prethodnog sloja. Uobičajeno je da ova vrsta slojeva dolazi na kraj neuronske mreže te obavlja ulogu klasifikatora gdje posljednji potpuno povezani sloj sadrži onoliko neurona koliko postoji izlaznih vrijednosti. Ukoliko je riječ o regresijskom problemu, obično se ne postavlja aktivacijska funkcija na izlazni sloj. Međutim, ako je riječ o klasifikacijskom problemu, koristi se *softmax* aktivacijska funkcija koja skalira izlazne vrijednosti tako da njihov zbroj bude jednak jedan.

U konačnici, jedna neuronska mreža za *end-to-end* klasifikaciju vremenskog signala bi se sastojala redom od ulaza, konvolucijskih slojeva, povratnih slojeva i potpuno povezanih slojeva. Konvolucijski slojevi najprije detektiraju intenzitet pojavljivanja određenih uzoraka koji se nalaze u signalu te se te vrijednosti prosljeđuju u povratni sloj. U njemu se detektiraju vremenske zavisnosti u signalu te kao konačni izlaz povratni sloj daje vektor vrijednosti. Slikovito rečeno, dobiveni vektor vrijednosti sadrži „sažetak” signala na temelju kojeg je moguće napraviti konačno predviđanje pomoću potpuno povezanog sloja.

3.2. Učenje

Znanje neuronske mreže je sadržano u njenim težinama ω , a njihove vrijednosti se određuju prilikom procesa učenja. Prije učenja se sve težine inicijaliziraju na nasumične vrijednosti te se kroz učenje one poboljšavaju i dovode do lokalnog optimuma. Kako bi učenje bilo moguće, najprije je potrebno odrediti funkciju, odnosno metriku, koja govori koliko kvalitetno neuronska mreža modelira zadani problem te se nju nastoji optimizirati učenjem. Ova funkcija se naziva funkcija gubitka (engl. *loss function*), a za potrebe ovog rada označiti ćemo ju sa oznakom L . Kao svoje argumente ona obično prima očekivane izlazne vrijednosti kojima neuronska mreža treba težiti i vrijednosti koje je neuronska mreža predvidjela. U principu se za regresijske probleme koristi prosječna kvadratna greška (engl. *root mean squared error*), dok se za klasifikacijske probleme koriste funkcije gubitka temeljene na entropiji kao što su binarna unakrsna entropija (engl. *binary crossentropy*) za binarnu klasifikaciju odnosno kategorična unakrsna entropija (engl. *categorical crossentropy*) za višeklasnu klasifikaciju.

Za podešavanje, odnosno ugađanje težina mreže se najčešće koriste tehnikama temeljenim na gradijentnom spustu koje se uglavnom sastoje od ponavljanja sljedećeg postupka. Najprije se podaci za učenje X propuste kroz neuronsku mrežu kako bi se dobila predviđanja \hat{y} . Ako točne vrijednosti označimo s y , može se izračunati gubitak pomoću funkcije gubitka $L(y, \hat{y})$. Zatim se pronalazi ovisnost gubitka o težinama mreže pomoću parcijalnih derivacija. Ukoliko je parcijalna derivacija u odnosu na težinu ω_i pozitivna, to znači da će se povećavanjem težine ω_i također povećati i vrijednost funkcije gubitka, a obrnuto vrijedi i za negativni slučaj. Budući da je cilj funkciju gubitka minimizirati, potrebno je dakle težinu smanjivati kada je parcijalna derivacija pozitivna, odnosno povećavati kada je negativna. Inkrementi u kojima se dešavaju promjene u težinama se reguliraju stopom učenja (engl. *learning rate*), a uobičajeno se koriste vrijednosti iz raspona [0.01, 0.0001].

Dodavanjem heuristika na osnovni gradijenti spust dobivaju se naprednije tehnike optimizacije težina, a nazivaju se optimizatorima. Među najkorištenijim optimizatorima su Adam (Adaptive moment estimation), SGD (Stochastic Gradient Descent) s momentom i RMSprop (Root Mean Square Propagation).

Uobičajen je postupak da se skup podataka podijeli na tri podskupa: skup za učenje, skup za validaciju i skup za testiranje. Primarna uloga skupa za učenje je da se na njemu provodi optimizacija težina dok se validacijski skup koristi za ugađanje hiperparametara modela (broj neurona u sloju, broj konvolucijskih filtera, itd.). Skup za testiranje se pritom ostavlja za sam kraj kako bi se na njemu mogao evaluirati model i dobiti što realističniju procjenu njegove točnosti.

Neuronske mreže su poznate po lakoći kojom mogu postati pretrenirane (engl. *overfitting*) odnosno preprilagođene skupu za učenje. Ova pojava se manifestira kroz lošu generalizaciju na nove, neviđene podatke. Smanjivanjem broja neurona se smanjuje kapacitet mreže te se na taj način i smanjuje mogućnost da dođe do pretreniranosti, ali ponekada to nije dovoljno pa je potrebno uvesti i druge tehnike regularizacije. Za potrebe ovog rada korištena je tehnika ranog zaustavljanja (engl. *early stopping*) i ispadanja (engl. *dropout*). Tehnika ranog zaustavljanja tijekom procesa učenja kontinuirano prati performanse modela na validacijskom skupu podataka. Trenutak kada se performanse na validacijskom skupu počnu pogoršavati ili stagnirati, a na skupu za učenje nastave poboljšavati označava da počinje nastupati pretreniranost jer je moć generalizacije na neviđenim podacima (koristeći validacijski skup podataka) opala. Mehanizam ranog zaustavljanja detektira ovakve situacije te u njima zaustavlja proces učenja.

Regularizacija putem ispadanja se sastoji od toga da se nasumični udio neurona „isključiti” prilikom svake epohe kako oni ne bi imali utjecaj na aktivacije narednih neurona. Također, vrijednosti njihovih težina se ne ažuriraju prilikom podešavanja težina te na taj način neuronska mreža mora naučiti temeljiti odluke na izlazima iz više neurona, umjesto iz pojedinačnih neurona. Na ovaj način se potiče da svi neuroni sudjeluju u donošenju odluke što pridonosi generalizacijskoj sposobnosti neuronske mreže.

3.3. Metrike vrednovanja

Kod klasifikacijskih modela u strojnom učenju, cilj je ispravno klasificirati ulazni primjerak u pripadajuću klasu. Klasifikacijski model predviđa klasu svake instance podataka, pripisujući svakoj instanci njezinu predviđenu oznaku. Ako se radi o binarnoj klasifikaciji, instance se svrstavaju u pozitivnu i negativnu klasu, a nakon što model klasificira sve instance, svaka će spadati u jedan od sljedeća četiri slučaja:

- stvarno pozitivne instance koje su točno predviđene kao pozitivne nazivaju se istinito pozitivne (TP),
- stvarno pozitivne instance koje su pogrešno predviđene kao negativne nazivaju se lažno negativne (FN),
- stvarno negativne instance koje su točno predviđene kao negativne nazivaju se istinito negativne (TN) i
- stvarno negativne instance koje su pogrešno predviđene kao pozitivne nazivaju se lažno pozitivne (FP).

Ova se raspodjela može predstaviti u tablici koja se naziva matrica zabune (engl. *confusion matrix*), koja u potpunosti opisuje ishod zadatka klasifikacije.

Tablica 2. Matrica zabune.

	Predviđeno pozitivne	Predviđeno negativne
Stvarno pozitivne	Istinito pozitivne (TP)	Lažno negativne (FN)
Stvarno negativne	Lažno pozitivne (FP)	Istinito negativne (TN)

3.3.1. Točnost

Budući da bi zasebna analiza sve četiri kategorije matrice zabune zahtijevala mnogo vremena, statističari su uveli korisne statističke vrijednosti koje mogu odmah opisati kvalitetu predviđanja s ciljem prenošenja strukture matrice zabune u jednu vrijednost. Za tu svrhu postoji točnost (engl. *accuracy*) koja predstavlja omjer između broja točno predviđenih instanci u odnosu na broj svih instanci u skupu podataka. Definirana je kao

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

Evaluacija modela ovom metrikom je vrlo česta u području strojnog učenja. Međutim, nije prikladna za ovaj rad jer bi zbog nebalansiranosti podataka mogla dati preoptimističnu ocjenu klasifikatora.

3.3.2. F-rezultat

Dvije metrike koje se koriste za mjerenje performansi modela su preciznost (engl. *precision*) i opoziv (engl. *recall*). Preciznost je udio primjeraka koje je model točno klasificirao kao pozitivne u odnosu na sve primjerke koje je klasificirao kao pozitivne, a opoziv je udio

primjeraka koje je model točno klasificirao kao pozitivne u odnosu na sveukupni broj pozitivnih primjeraka.

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

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

F -rezultat (engl. F -score) je definiran kao ponderirani prosjek preciznosti i opoziva ovisno o težinskoj funkciji β . F_1 -rezultat mjeri harmonijsku sredinu između preciznosti i opoziva. F_1 -rezultat može imati različite indekse koji daju različite težine preciznosti i opoziva.

$$F_\beta = (1 + \beta^2) * \frac{P * R}{\beta^2 * P + R}$$

Ako uzmemo da je $\beta = 1$, standardni F_1 -rezultat je

$$F_1 = 2 * \frac{P * R}{P + R}$$

Po definiciji, točnost je definirana za svaku matricu zabune M i kreće se u stvarnom jediničnom intervalu $[0, 1]$. Najbolja vrijednost točnosti iznosi jedan i to odgovara savršeno točnoj klasifikaciji $M = \begin{pmatrix} n_+ & 0 \\ 0 & n_- \end{pmatrix}$, dok najgora vrijednost točnosti iznosi nula i to odgovara savršeno pogrešnoj klasifikaciji $M = \begin{pmatrix} 0 & n_+ \\ n_- & 0 \end{pmatrix}$ gdje je n_+ broj pozitivnih instanci, a n_- broj negativnih instanci u skupu podataka.

3.3.3. Matthewsov koeficijent korelacije

Matthewsov koeficijent korelacije (MCC) (engl. *Matthews Correlation Coefficient*) je alternativna mjera na koju ne utječe problem neuravnoteženih skupova podataka. MCC je metoda matrice nepredviđenih okolnosti za izračun Pearsonovog proizvod-moment koeficijenta korelacije (engl. *Pearson product-moment correlation coefficient*) između stvarnih i predviđenih vrijednosti. Formula za MCC je

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}}$$

MCC je jedina stopa klasifikacije koja generira visoku ocjenu samo ako je prediktor uspio točno predvidjeti većinu instanci pozitivnih podataka i većinu instanci negativnih podataka. Kreće se u intervalu $[-1, 1]$, s ekstremnim vrijednostima -1 i 1 postignutim u slučaju savršeno pogrešne klasifikacije, odnosno savršene klasifikacije, dok je $MCC = 0$ očekivana vrijednost za klasifikator kada je u pitanju problem bacanja novčića.

3.3.4. Površina ispod krivulje radne karakteristike prijammnika

Površina ispod krivulje radne karakteristike prijammnika (ROC-AUC) (engl. *Receiver Operating Characteristic-Area Under Curve*) predstavlja površinu ispod ROC krivulje te ima veću vrijednost što je binarni klasifikator bolji. Stoga je za izračunati ovu metriku najprije potrebno konstruirati ROC krivulju i zatim izračunati površinu ispod nje. ROC krivulja na svojoj apscisi prikazuje stopu pravih pozitivna (TPR) (engl. *True Positive Rate*), a na ordinati stopu lažnih pozitivna (FPR) (engl. *False Positive Rate*), a one su definirane na slijedeći način:

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

$$FPR = \frac{FP}{FP + TN}$$

Binarni klasifikator temeljen na neuronskoj mreži kao svoj izlaz daje vjerojatnost da instanca pripada pozitivnoj klasi. Variranjem granice odluke se mijenja i predviđena klasa za instance te se time i mijenjaju TPR i FPR metrike. Crtanjem dobivenih (FPR, TPR) parova se dobiva ROC krivulja. Računanjem površine ispod nje se dobiva ROC-AUC metrika binarnog klasifikatora.

Kako ova metrika nije izravno primjenjiva na višeklasne klasifikatore, primjenjuje se tehnika jedan protiv svih (engl. *one-vs-all*). Ova tehnika se sastoji od toga se odredi ROC-AUC za svaku klasu zasebno, a zatim se izračuna prosječna vrijednost koja predstavlja vrijednost za cijeli klasifikator. Prilikom razmatranja svake klase, razmatranu klasu se smatra pozitivnom dok se ostale smatra negativnom klasom te se na taj način dobiva problem binarne klasifikacije. Može se zaključiti da ako višeklasni model predviđa k klasa, tada je potrebno postupak jedan protiv svih ponoviti k puta i to za svaku klasu posebno.

3.4. Učenje prijenosom znanja

Duboko učenje je kroz prijašnje godine privuklo sve veću pozornost istraživača i uspješno je primijenjeno na brojne situacije u stvarnom svijetu. Algoritmi dubokog učenja pokušavaju naučiti značajke visoke razine iz velike količine podataka. Ovakvo učenje, za razliku od tradicionalnog strojnog učenja, može automatski pronaći značajke iz podataka nadziranim ili polu nadziranim algoritmom učenja. Nasuprot tome, kod tradicionalnih metoda strojnog učenja značajke se moraju ručno dizajnirati što korisnicima ozbiljno otežava dobivanje dobrih rezultata. Može se reći da je duboko učenje algoritam za učenje temeljeno na velikim podacima u strojnom učenju.

Ovisnost o podacima jedan je od najozbiljnijih problema dubokog učenja. Duboko učenje ima vrlo čvrstu ovisnost o velikoj količini podataka za učenje u usporedbi s tradicionalnim metodama strojnog učenja, jer mu je potrebna velika količina podataka za razumijevanje latentnih obrazaca podataka. Može se uočiti zanimljiv fenomen gdje su veličina modela i veličina potrebne količine podataka u gotovo linearnom odnosu. Prihvatljivo objašnjenje je da za određeni problem izražajni prostor modela mora biti dovoljno velik da se otkriju uzorci ispod podataka. Prvi slojevi u modelu mogu identificirati značajke visoke razine iz podataka za učenje, a sljedeći slojevi mogu identificirati informacije potrebne za donošenje konačne odluke.

Nedovoljno podataka za učenje je neizbježan problem u nekim posebnim domenama. Prikupljanje podataka je složeno i skupo, zbog čega je iznimno teško izgraditi skup podataka velikih razmjera visoke kvalitete. Na primjer, svaki uzorak u bioinformatičkom skupu podataka često zahtjeva provedbu kliničkog ispitivanja nad određenoj količini pacijenta. Osim toga, podaci mogu vrlo lako zastarjeti i zbog toga se ne mogu učinkovito primijeniti u novim zadacima.

Učenje prijenosom znanja (engl. *transfer learning*) ublažava hipotezu da podaci za učenje moraju biti neovisni i identično raspodijeljeni s testnim podacima, što nas motivira da koristimo prijenos znanja gdje se susretnemo s problemom nedovoljne količine podataka za učenje. U učenju gdje prenosimo znanje, podaci za učenje i testiranje ne moraju biti neovisni i identično raspodijeljeni, a model u ciljnoj domeni ne mora krenuti učiti od nule, što može značajno smanjiti potrebnu količinu podataka i skratiti vrijeme potrebno za učenje modela u ciljnoj domeni.

3.5. Unakrsna provjera valjanosti

Unakrsna provjera valjanosti (engl. *cross-validation*) je metoda ponovnog uzorkovanja podataka za procjenu sposobnosti generalizacije prediktivnih modela i sprječavanje pretreniranja [8]. Unakrsna provjera valjanosti pripada obitelji Monte Carlo metoda.

Razmotrimo skup podataka D koji se sastoji od n označenih instanci (tj. slučajeva), $x_i, i = 1 \dots n$. Svaki slučaj je opisan skupom atributa (tj. značajki). Pretpostavimo da svaki slučaj x_i pripada točno jednoj klasi y_i . Tipičan primjer iz bioinformatike je skup podataka o ekspresiji gena temeljen na podacima mikroniza DNK, gdje svaki slučaj predstavlja jedan označeni uzorak tumora opisan profilom ekspresije gena. Jedan od uobičajenih izazova odnosi se na razvoj klasifikatora koji može pouzdano predvidjeti klasu novih, neviđenih uzoraka tumora na temelju njihovih profila ekspresije. Konceptualno, prediktivni model, $f()$, je pravilo za dodjeljivanje oznake klase slučaju na temelju skupa podataka D , tj. $f(x, D) = \hat{y}_i$, gdje je \hat{y}_i predviđena oznaka klase za slučaj x_i . U strojnom učenju, konstrukcija takvog modela zove se nadzirano učenje.

Središnje pitanje u nadziranom učenju odnosi se na točnost rezultirajućeg modela. Ovdje je ključni problem prekomjerno prilagođavanje modela [9]. Vrlo je jednostavno izgraditi model koji je savršeno prilagođen odabranom skupu podataka, ali onda se ne može dobro generalizirati na nove, neviđene podatke. Za primjer uzmimo problem univarijatne regresije gdje želimo predvidjeti zavisnu varijablu y iz nezavisne varijable x na temelju n opažanja $(x_i, y_i), i = 1 \dots n$. Mogli bismo upotrijebiti polinom stupnja $n - 1$ da krivulju savršeno uklopimo kroz te točke i zatim upotrijebiti krivulju za ekstrapolaciju vrijednosti y_{i+1} za novi slučaj, x_{i+1} . Međutim, ova će krivulja vrlo vjerojatno biti previše prilagođena dostupnim podacima te ne samo da će odražavati odnos između zavisne i nezavisne varijable, već će model također imati grešku naučenu kroz inherentni šum u skupu podataka. S druge strane, koristeći jednostavniji model kao što je linija najmanjih kvadrata, doći će do manjeg utjecaja inherentnog šuma iz podataka, ali možda neće dobro obuhvatiti odnos između varijabli. Za takav se model kaže da je nedovoljno prilagođen (engl. *underfitting*). Ako je model prekomjerno prilagođen ili pak nedovoljno prilagođen, predikcije modela koristeći dosad neviđene podatke neće biti zadovoljavajuće što znači da takvi modeli ne generaliziraju dovoljno. Pronaći pravu ravnotežu prilagođenosti modela u svrhu dobivanja što bolje generaliziranog modela je veliki izazov.

Kako možemo procijeniti sposobnost generalizacije modela? U idealnom slučaju, procijenili bismo model koristeći nove podatke koji potječu iz iste populacije kao i podaci koje smo koristili za izradu modela. Međutim, u praksi nove neovisne validacijske studije često nisu izvedive. Također, prije nego što uložimo vrijeme i druge resurse za daljnju provjeru valjanosti, preporučljivo je prvo procijeniti prediktivnu točnost modela. To se obično radi metodama ponovnog uzorkovanja podataka, kao što je unakrsna provjera valjanosti.

Skup podataka koji je dostupan za izradu i procjenu prediktivnog modela naziva se skup za učenje, D_{learn} . Pretpostavlja se da je ovaj skup podataka uzorak iz populacije od interesa. Metode nasumičnog poduzorkovanja koriste se za generiranje skupova za učenje, D_{train} , i skupova za testiranje, D_{test} , koji se generiraju iz skupa za učenje. Model se zatim uči pomoću skupova za učenje i testira na skupovima za testiranje. Različite metode nasumičnog poduzorkovanja razlikuju se s obzirom na način na koji se generiraju skupovi za učenje i testiranje.

Izraz „učenje“ podrazumijeva da algoritam učenja primjenjujemo na podskup podataka. Rezultirajući model, $\hat{f}(x, D_{train})$, samo je procjena konačnog modela koji proizlazi iz primjene iste funkcije učenja na cijeli skup učenja, $f(x, D_{learn})$. Evaluacija modela temeljena na ponovljenom poduzorkovanju znači da se funkcija učenja primjenjuje na nekoliko podskupova

podataka, a rezultirajući modeli, \hat{f}_j , naknadno se procjenjuju na drugim podskupovima (tj. skupovi za testiranje ili validaciju), koji nisu korišteni tijekom učenja modela. Prosjek izvedbe koju modeli postižu na tim podskupovima je procjena izvedbe konačnog modela, $f(x, D_{learn})$.

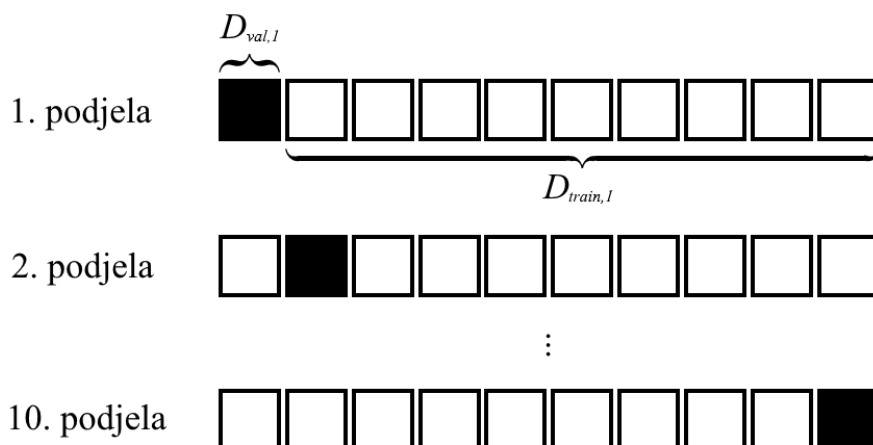
Pretpostavimo da je svakom slučaju pridružena točno jedna ciljna oznaka y_i . U slučaju klasifikacije, y_i je diskretna oznaka klase. Klasifikator je poseban slučaj prediktivnog modela koji instanci dodjeljuje diskretnu oznaku klase. U zadacima regresije, cilj je obično vrijednost u realnom skupu brojeva, $y_i \in R$. Prediktivni model $f()$ procjenjuje cilj y_i slučaja x_i kao $f(x_i) = \hat{y}_i$. Funkcija gubitka, $L(y_i, \hat{y}_i)$, kvantificira pogrešku predikcije. Na primjer, korištenjem funkcije gubitka 0-1 u zadatku klasifikacije, gubitak je 1 ako je $y_i \neq \hat{y}_i$ i 0 u suprotnom. S funkcijom gubitka sada možemo izračunati dvije različite pogreške: pogrešku učenja i pogrešku testiranja.

Pogreška učenja govori nam o prilagodbi na skupove za učenje, dok je pogreška testiranja procjena stvarne pogreške u predviđanju. Ova procjena kvantificira sposobnost generalizacije modela. Također, pogreška učenja ima tendenciju podcijeniti pravu pogrešku predviđanja, jer se isti podaci koji su korišteni za učenje modela ponovno koriste za procjenu točnosti modela.

Među različitim strategijama ponovnog uzorkovanja podataka, jedna od najjednostavnijih je metoda jednog zadržavanja, koja nasumično odabire neke slučajeve iz skupa podataka koji će tvoriti skup za testiranje, dok preostali slučajevi čine skup za učenje. Često skup za testiranje sadrži oko 10% do 30% dostupnih slučajeva, a skup za učenje sadrži oko 90% do 70% slučajeva. Ako je skup podataka dovoljno velik, i posljedično, ako su i skupovi za učenje i skupovi za testiranje veliki, tada promatrana pogreška testiranja može biti pouzdana procjena stvarne pogreške modela za nove, neviđene slučajeve.

Kod k -strukog nasumičnog poduzorkovanja (engl. *k-fold random subsampling*), metoda jednog zadržavanja ponavlja se k puta, tako da se generira k parova skupova za učenje, $D_{train,j}$, i skupova za testiranje, $D_{test,j}$, $j = 1 \dots k$. Funkcija učenja se primjenjuje na svaki skup za učenje, a dobiveni model zatim se primjenjuje na odgovarajući skup za testiranje. Točnost modela je procijenjena kao prosjek za svih k skupova za testiranje. Bilo koji par skupa za učenje i testiranje su disjunktivni, tj. skupovi nemaju nijedan zajednički slučaj, $D_{train,j} \cap D_{test,j} = \emptyset$. Međutim, ako gledamo bilo koja dva skupa za učenje ili dva skupa za testiranje, oni mogu imati preklapanja u podacima.

Unakrsna provjera valjanosti slična je metodi k -strukog nasumičnog poduzorkovanja, no ovdje se uzorkovanje provodi na takav način da se nijedna dva skupa za testiranje ne preklapaju. U k -strukoj unakrsnoj provjeri valjanosti (engl. *k-fold cross-validation*), dostupni skup za učenje je podijeljen na k disjunktivnih podskupova približno jednake veličine. Ovdje se „poklapanje“ odnosi na broj rezultirajućih podskupova. Ova se particija izvodi nasumičnim uzorkovanjem slučajeva iz skupa za učenje bez zamjene. Model se uči pomoću $k - 1$ podskupova, koji zajedno predstavljaju skup za učenje. Zatim se model primjenjuje na preostali podskup, koji se označava kao skup za provjeru valjanosti, te se mjeri točnost modela. Ovaj postupak se ponavlja sve dok svaki od k podskupova ne posluži kao skup za validaciju. Prosjek k mjerenja točnosti na k skupovima za validaciju je unakrsno validirana točnost modela. Slika 3. ilustrira ovaj proces za $k = 10$, tj. 10-struku unakrsnu provjeru valjanosti. U prvoj podijeli, prvi podskup služi kao skup za validaciju, $D_{val,1}$, a preostalih devet podskupova služi kao skup za učenje, $D_{train,1}$. U drugoj podijeli, drugi podskup je skup za validaciju, a preostali podskupovi čine skup za učenje, i tako dalje, sve dok se svi podskupovi ne iskoriste za validaciju.



Slika 3. 10-struka unakrsna provjera valjanosti.

Unakrsno validirana točnost je, na primjer, prosjek svih deset točnosti postignutih na skupovima za validaciju. Općenitije, neka \hat{f}_{-k} označava model koji je naučen koristeći sve osim k -tog podskupa skupa za učenje. Vrijednost $\hat{y}_i = \hat{f}_{-k}(x_i)$ je predviđena ili procijenjena vrijednost za oznaku stvarne klase, y_i , slučaja x_i , koji je element k -tog podskupa. Unakrsno validirana procjena pogreške predviđanja, \hat{E}_{cv} , tada je

$$\hat{E}_{cv} = \frac{1}{n} \sum_{i=1}^n L(y_i, \hat{f}_{-k}(x_i))$$

Unakrsna provjera valjanosti često uključuje stratificirano slučajno uzorkovanje, što znači da se uzorkovanje izvodi na takav način da omjeri razreda u pojedinačnim podskupovima odražavaju omjere u skupu za učenje.

4. Skupovi podataka

U ovom su poglavlju opisani skupovi podataka koji su korišteni za učenje prijenosom znanja u ovom istraživanju. Oba skupa podataka sadrže zvučne datoteke, Speech Commands je skup podataka izgovorenih riječi (akustični signali), a RAVDESS je skup izgovorenih rečenica u određenoj emociji. Fokus ovog istraživanja bio je na učenje prijenosom znanja unutar iste domene (klasifikacija zvuka).

4.1. Speech Commands

Ovaj skup podataka sadrži zvučne datoteke izgovorenih riječi dizajniranih da pomognu pri učenju i procjeni sustava za uočavanje ključnih riječi. Sastavio ga je i objavio Google Brain [10].

Postoje dvije verzije Speech Commands skupa podataka, prva verzija (v0.01) ima 65 000 izgovora riječi različitih govornika. Svaki audio zapis traje jednu sekundu i pripada jednoj od 30 klasa koje odgovaraju uobičajenim riječima engleskog jezika kao što su *Yes, No, Go, Stop, Left, Down*, brojčane znamenke, itd. Druga verzija (v0.02) ima 105 000 izgovora riječi spremljenih u WAV formatu, a korištena frekvencija uzorkovanja iznosi 16 kHz. Svaki audio zapis traje oko jednu sekundu i pripada jednoj od 35 klasa, a osim toga postoje i neke duže snimke glasova koji ne sadrže govor. U literaturi se može pronaći više informacija o njima i upute o tome gdje ih se može preuzeti [10]. U ovom eksperimentu je korištena verzija v0.02, koja sadrži veći broj audio zapisa izgovora više različitih riječi (tj. klasa) od prethodne verzije.

U pretprocesiranju ovog skupa podataka se svi audio zapisi nadopunjavaju bijelim šumom tako da imaju isto trajanje. Njihova duljina određena je duljinom najduljeg audio zapisa u skupu podataka. Bijeli šum sadržan je u samom skupu podataka. Također, ponovnim uzorkovanjem frekvencija uzorkovanja audio zapisa je smanjena na 8 kHz kako bi rezultati bili što bolji. U literaturi je utvrđeno kako je ova frekvencija uzorkovanja pogodna za učenje prijenosom znanja na ovome skupu podataka [11].

4.2. RAVDESS

Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) je skup podataka koji je rodno uravnotežen te sadrži 7 356 zvučnih datoteka emocionalnog govora i pjesme 24 profesionalna glumca, koji vokaliziraju leksički podudarne izjave neutralnim sjevernoameričkim naglaskom. Govor uključuje mirne, sretne, tužne, ljutite, uplašene izraze, izraze iznenađenja i gađenja, a pjesma sadrži mirne, sretne, tužne, ljutite i strahovite emocije. Svaki izraz govora se proizveo na dvije razine emocionalnog intenziteta, uz dodatni neutralni izraz. Izgovori su ocijenjeni 10 puta na emocionalnu valjanost, intenzitet i autentičnost. Zvučni zapisi su snimljeni frekvencijom uzorkovanja od 48 kHz. U literaturi se može pronaći više informacija o ovom skupu podataka i upute o tome gdje se može preuzeti [12].

U pretprocesiranju ovog skupa podataka se također svi audio zapisi nadopunjavaju bijelim šumom tako da imaju isto trajanje te je njihova duljina također određena duljinom najduljeg audio zapisa u skupu podataka. Korišten je bijeli šum koji je sadržan u Speech Commands skupu podataka.

4.3. Usporedba skupova podataka

Tablica 3. Usporedba skupova podataka.

Ime skupa podataka	Vrsta problema	Broj klasa	Broj primjeraka	Izvorna frekvencija uzorkovanja
--------------------	----------------	------------	-----------------	---------------------------------

Speech Commands	Klasifikacija	35	105 000	16 kHz
RAVDESS	Klasifikacija	8	7 356	48 kHz

4.4. Standardizacija podataka

Preprocesiranje podataka je ključan korak prije korištenja bilo kakvih algoritama istraživanja nad podacima kako bi što više poboljšali rezultat modela. Normalizacija skupa podataka jedan je od procesa preprocesiranja podataka, u kojem se podaci skaliraju tako da „padnu“ u mali određeni raspon. Normalizacija prije klasifikacije posebno je potrebna za metriku udaljenosti, poput euklidske udaljenosti koje su osjetljive na varijacije unutar veličine ili mjerila značajke. U stvarnim primjenama, zbog varijacija u odabiru vrijednosti značajki, jedna značajka može nadjačati drugu. Normalizacija sprječava nadmašivanje značajki koje imaju velike apsolutne vrijednosti nad značajkama s manjim apsolutnim vrijednostima. Cilj bi bio izjednačiti dimenzije ili veličinu, a također i varijabilnost tih značajki. Svođenje značajki na istu skalu ubrzava konvergenciju modela.

Tehnike predobrade podataka primjenjuju se na neobrađene podatke kako bi podaci bili dosljedni, čisti i bez šuma. Normalizacija podataka standardizira neobrađene podatke pretvarajući ih u određeni raspon pomoću linearne transformacije koja može generirati kvalitetnu grupaciju i poboljšati točnost algoritama klasifikacije. Ne postoji univerzalno pravilo za normalizaciju skupova podataka i stoga je izbor određenog pravila normalizacije u velikoj mjeri prepušten slobodnom izboru. Metode normalizacije podataka uključuju *Z*-rezultat (engl. *Z-score*), Min-Max i decimalno skaliranje. U *Z*-rezultatu vrijednost x se standardizira na temelju srednje vrijednosti μ skupa podataka i standardne devijacije σ skupa podataka, a ova metoda je korisna kada su stvarni minimum i maksimum nepoznati. Decimalno skaliranje standardizira podatke pomicanjem decimalne točke vrijednosti x , a broj pomaknutih decimalnih točaka ovisi o maksimalnoj apsolutnoj vrijednosti atributa. Min-Max transformira skup podataka između 0.0 i 1.0 oduzimanjem minimalne vrijednosti od svake vrijednosti podijeljene s rasponom vrijednosti za svaku pojedinačnu vrijednost.

U ovom radu korištena je *Z*-rezultat transformacija. To je oblik standardizacije koji se koristi za transformaciju normalnog rezultata u standardni rezultat. Formula standardizacije *Z*-rezultata definirana je kao

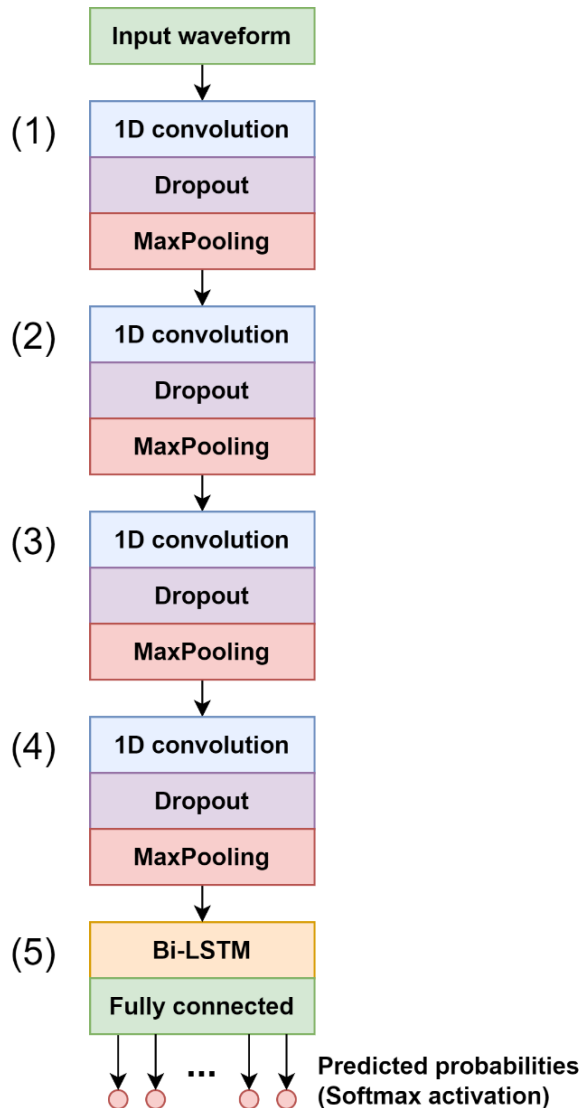
$$z = \frac{x - \mu}{\sigma}$$

Standardizacija ovom transformacijom uključuje uklanjanje prosjeka kako bi podaci bili centrirani oko nule i dijeljenje sa standardnom devijacijom kako bi se dobila jedinična varijanca i time „suzila“ odnosno „proširila“ distribucija vrijednosti prema potrebi. Jedno važno ograničenje standardizacije *Z*-rezultata je da se mora primijeniti kao standardizacija cijelog skupa podataka, a ne kao standardizacija samo unutar klase podataka.

5. Analiza podataka i rezultati

5.1. Postavke eksperimenta

Model se sastoji od ulaznog sloja, četiri konvolucijska sloja gdje nakon svakog slijedi sloj izbacivanja (engl. *dropout*) te sloj sažimanja maksimumom (engl. *max pooling*). Slika 4. prikazuje slojeve modela. Na kraju se nalaze povratni LSTM sloj, potpuno povezani sloj (engl. *dense*) i izlazni sloj. Sveukupni broj neurona iznosi 146 632.



Slika 4. Prikaz slojeva modela.

5.1.1. Aktivacijska funkcija

Kako bi model mogao modelirati nelinearnosti, bilo je potrebno primijeniti nelinearne aktivacijske funkcije. Izlazni neuroni modela koriste softmax aktivacijsku funkciju koja je u biti sastavljena od više sigmoidnih funkcija čiji je zbroj jednak jedan. Sigmoidna funkcija vraća vrijednosti u rasponu od 0 do 1 te se te vrijednosti mogu tretirati kao vjerojatnosti podatkovnih točaka određene klase. Ova aktivacijska funkcija se može koristiti za probleme višeklasne klasifikacije za razliku od sigmoidnih funkcija koje se koriste za binarnu klasifikaciju. Softmax funkcija za svaku podatkovnu točku svih pojedinačnih klasa vraća vjerojatnost te se može izraziti kao

$$\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}}, i = 1 \dots k$$

Za bolje rezultate modela potrebno je uzeti u obzir mnogo stvari poput broja skrivenih slojeva u mreži, metodu učenja, podešavanje hiperparametara itd. Funkcija aktivacije neurona jedan je od najvažnijih parametara koje treba uzeti u obzir. Odabir prave aktivacijske funkcije za bilo koji zadatak može biti naporan proces i može zahtijevati puno istraživanja. Ne postoji pravilo za odabir bilo koje funkcije aktivacije, ali njen izbor ovisi o zadatku koji treba riješiti. Različite aktivacijske funkcije imaju svoje prednosti i nedostatke, a to ovisi o vrsti problema kojeg želimo riješiti. Na primjer, za probleme klasifikacije, kombinacija sigmoidnih funkcija daje bolje rezultate. Ako se suočimo s problemom nestajanja gradijenta, tj. gradijenta koji dostiže vrijednost nula, tada se izbjegavaju sigmoidna i tanh funkcija. ReLU funkcija je najčešće korištena funkcija i u većini slučajeva radi bolje od ostalih aktivacijskih funkcija. Ona se mora koristiti samo u skrivenim slojevima, a ne u vanjskom sloju. Ako u našoj mreži ima „mrtvih“ neurona, tada možemo koristiti Leaky ReLU funkciju. Studije su pokazale da sigmoidna i tanh funkcija nisu prikladne za skrivene slojeve jer nagib funkcije postaje vrlo mali kako unos postaje vrlo velik ili vrlo malen, što zauzvrat usporava spuštanje gradijenta. ReLU je najpoželjniji izbor za primjenu sa skrivenim slojevima budući da je derivat ReLU funkcije 1. Također, Leaky ReLU se može koristiti u slučaju nultih derivata. Cilj je odabrati funkciju aktivacije koja će brže aproksimirati funkciju i koja može omogućiti brže učenje modela. U ovom je eksperimentu za skrivene slojeve korištena ReLU aktivacijska funkcija.

5.1.2. Funkcija gubitaka

Uz softmax aktivacijsku funkciju, pridružena je kategorička funkcija unakrsnog entropijskog gubitka (engl. *categorical cross-entropy*). Kada se sloj aktivira sa softmax funkcijom, zbroj izlaza iznosi 1, a tumačenje je da svaki od k izlaza procjenjuje vjerojatnost ili pouzdanost da instanca pripada određenoj klasi. Na primjer, pretpostavimo da postoje tri klase slika: psi, mačke i sve ostalo. Izlaz $[0.7, 0.2, 0.1]$ tumačio bi se kao da postoji 70% šanse da se radi o slici psa, 20% šanse da je to slika mačke i 10% šanse da je to slika nečeg drugog. Međutim, kategorička funkcija unakrsnog entropijskog gubitka kažnjava samo 30% pouzdanosti da slika ne prikazuje psa (vjerojatno lažno negativno). Ne kažnjava 20% šanse za mačku (vjerojatno lažno pozitivno). Argument u korist ovakvog pristupa je taj da, budući da je funkcija negativnog logaritamskog gubitka konkavna, ponderiranje od $[0.7, 0.2, 0.1]$ ima veći gubitak od $[0.7, 0.1, 0.2]$, unatoč tome što su oba rezultata identična. Stoga se vjerojatno lažno negativne vrijednosti trebaju zanemariti kako bi gubici bili identični. U osnovi, funkcija gubitka kaže da nije bitno što softmax predviđa za negativne klase, važno je samo da predviđa 70% za pozitivnu klasu. Posljedica ovakvog pristupa je da tipična funkcija gubitaka primijenjena na softmax ne može izravno kazniti vjerojatno lažno pozitivne rezultate.

Softmax je generalizacija binarne klasifikacije, međutim, detalji implementacije se razlikuju. U binarnoj klasifikaciji obično postoji samo jedan izlaz koji predviđa vjerojatnost pozitivnog rezultata. Ne postoji drugi izlaz koji predviđa vjerojatnost negativnog rezultata. Za slučaj kategoričke klasifikacije s jednom oznakom (tj. softmax aktivacija) standardni kategorički unakrsni entropijski gubitak je

$$L_{cce} = -\frac{1}{M} \sum_{k=1}^K \sum_{m=1}^M y_m^k * \log(h_{\theta}(x_m, k))$$

gdje je M broj instanci za učenje, K broj klasa, y_m^k ciljna oznaka za instancu m koja pripada klasi k , x unos za primjer instance m i h_{θ} model s težinama neuronske mreže θ .

5.1.3. Optimizator

Učenje u neuronskim mrežama provodi se minimiziranjem funkcije pogreške koja se još naziva i funkcija gubitka. Ova funkcija stoga mjeri razliku između očekivanih izlaza i onih izračunatih koristeći instancu podataka. Pogreška blizu nule implicira da neuronska mreža ispravno klasificira podatke na temelju kojih je naučila. Međutim, određivanje globalnog minimuma postaje sve teže kako se veličina mreže povećava, a to je u praksi irelevantno jer globalni minimum često dovodi do prekomjernog prilagođavanja. Dakle, cilj je minimizirati grešku, ali i pritom paziti da ne dođe do pretreniranosti modela na podacima za učenje. Idealno je naučiti mrežu da može dobro klasificirati podatke za učenje, ali i podatke koje nikada prije nije vidjela (podaci za validaciju).

Adam je algoritam prvog reda gradijenta baziran na stohastičkim ciljnim funkcijama, temeljen na adaptivnim procjenama momenata nižeg reda [13]. Prvi trenutak normaliziran drugim trenutkom daje smjer ažuriranja. Adamova ažuriranja izravno se procjenjuju korištenjem tekućeg prosjeka prvog i drugog trenutka gradijenta. Ovaj algoritam izračunava prilagodljive stope učenja za svaki parametar. Uz pohranjivanje eksponencijalno padajućeg prosjeka prošlih kvadratnih gradijenata v_t poput AdaDelta [14] i RMSprop [15], Adam također čuva eksponencijalno opadajući prosjek prošlih gradijenata m_t , slično momentu.

Tijekom posljednjih godina Adam optimizator je postao jedna od najčešće korištenih optimizacijskih metoda za učenje neuronskih mreža. Zbog toga je korišten Adam optimizator, a za stopu učenja korištena vrijednost iznosi 0.001.

5.1.4. Unakrsna provjera valjanosti

Nasumično uzorkovanje općenito je ispravan odabir ako je izvorni skup podataka dovoljno velik, no ako nije, uvodi se pristranost zbog pogreške uzorkovanja. Stratificirano uzorkovanje je metoda uzorkovanja koja smanjuje pogrešku uzorkovanja u slučajevima kada se populacija može podijeliti u podskupine. Stratificirano uzorkovanje provodimo dijeljenjem populacije u homogene podskupine, koje se nazivaju stratumi, a zatim primjenjujemo jednostavno slučajno uzorkovanje unutar svake podskupine. Kao rezultat toga, skup podataka za testiranje je reprezentativan za populaciju, budući da je postotak svakog stratuma sačuvan. Slojevi trebaju biti razdvojeni, tj. svaki element unutar populacije mora pripadati jednom i samo jednom stratumu.

U svrhu smanjivanja ovakve vrste pogreške, u ovom eksperimentu je korištena stratificirana 4-struka unakrsna provjera valjanosti kako bi se mogla procijeniti sposobnost generalizacije rezultirajućeg modela.

Osim korištenja stratificirane k -struke unakrsne provjere valjanosti, taj postupak se ponovio tri puta kako bi se sakupio dovoljan broj mjerenja za statistički test. Time je naučeno 12 modela gdje svaki od njih predstavlja određenu konfiguraciju eksperimenta.

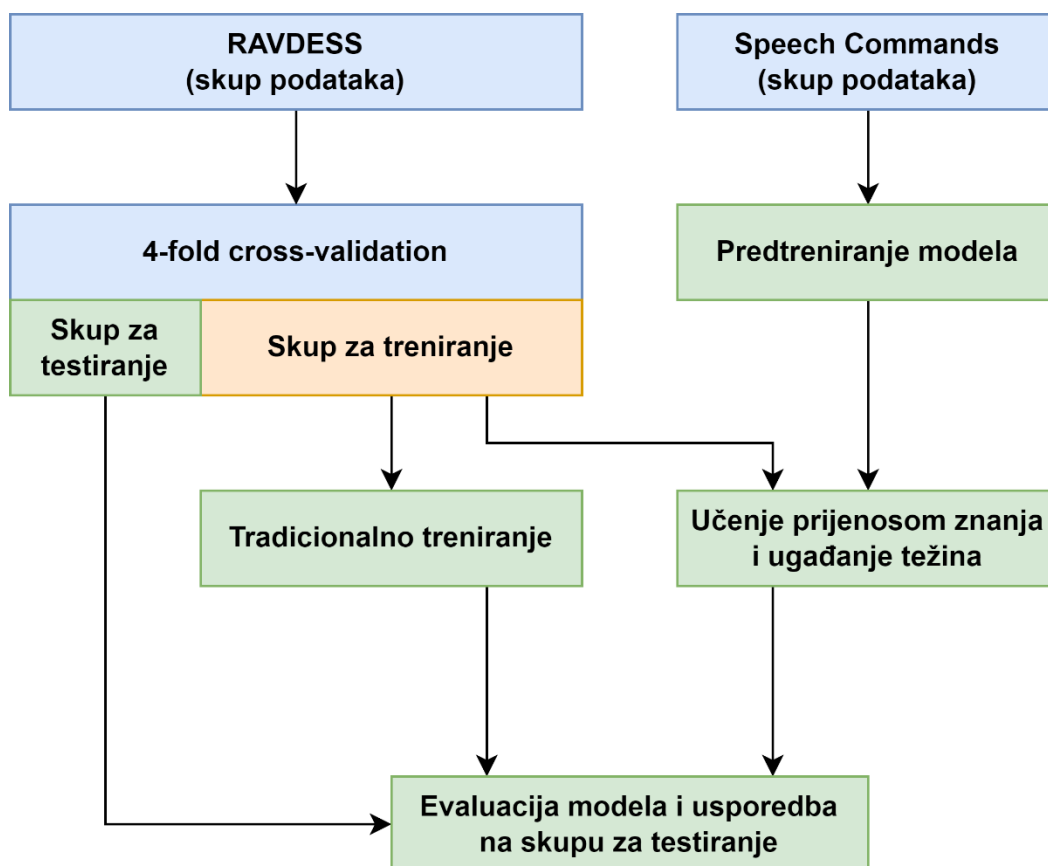
5.2. Proces učenja prijenosom znanja

Kvaliteta rezultata dobivenih prijenosom znanja osim što ovisi o kvaliteti prethodno naučenog modela, ovisi i o tome koje se znanje prenosi te na koji način.

Za odgovoriti na prvo pitanje, u ovome radu je ispitan prijenos znanja za prvih n blokova sa slike 4. Za n su ispitane vrijednosti u rasponu [1, 5] što predstavlja sve mogućnosti. Naime, poznato je kako rani konvolucijski slojevi uče prepoznavati jednostavne oblike, a svaki idući dublji sloj uči kombinirati oblike prethodnog sloja kako bi dobio kompleksnije oblike. Zbog toga ima smisla prenositi prvih n slojeva odnosno ne bi imalo smisla preskočiti neki raniji sloj budući da se znanje kasnijih slojeva temelji na ranijem sloju. Može se reći kako znanje iz svakog bloka predstavlja osnovu za znanje u idućem bloku (Slika 4). Intuicija prijenosa znanja se

temelji na tome da postoji neko zajedničko znanje koje može poslužiti za rješavanje izvorišnog i odredišnog zadatka. U sekvencijalnom modelu kao što je na slici 4, zajedničko znanje se nalazi u prvih n blokova te nakon toga znanje postaje specifično za problem koji se rješava. Nije moguće unaprijed odrediti u kojem bloku dolazi do specijalizacije znanja te je zbog toga bilo potrebno ispitati sve mogućnosti kako bi se otkrila ona koja daje najbolje rezultate. Prijenos znanja podrazumijeva prenošenje odabranih težina, dok se ostale težine u modelu nasumično inicijaliziraju.

Drugo pitanje se odnosi na način na koji se provodi učenje prijenosom znanja. Uobičajena je praksa da se znanje prenosi u dva koraka. U prvome koraku se prenesene težine zamrznu te se ostale težine ugađaju na odredišnom zadatku, a potom kada model konvergira nastavlja se podešavanje svih težina. Zamrzavanje težina na početku je potrebno kako bi one ostale sačuvane dok se ostale težine dovedu u pogodno stanje. Kada one ne bi bile zamrznute, moglo bi se dogoditi da dođe do gubitka znanja u njima dok se ostale težine ugađaju iz svojih nasumičnih u optimalne vrijednosti. Tada je moguće odmrznuti težine te provesti dodatno ugađanje težina.



Slika 5. Proces učenja prijenosom znanja.

Shematski prikaz postupka cijelog eksperimenta može se vidjeti na slici 5. Eksperiment započinje tako da se najprije generiraju podskupovi RAVDESS skupa podataka. Na skupu za učenje se provodi tradicionalno učenje modela bez prijenosa znanja, a zatim se vrši validacija njegove točnosti na skupu za testiranje. Slično tome, model temeljen na prijenosu znanja najprije se uči na Speech Commands skupu podataka, a zatim se novi model uči koristeći preneseno znanje iz prethodno naučenog modela i vrši validacija na istim podskupovima kao i

tradicionalni model. Metrike se bilježe tijekom cijelog procesa unakrsne validacije te se u konačnici izračunava prosječna vrijednost, standardna devijacija i provodi statistički test.

5.3. Rezultati

Rezultati dobiveni metodologijom sa slike 5. su prikazani u tablici 4. Za svaku metriku je podebljana maksimalna vrijednost, a metrike u kojima postoji statistički značajna razlika između učenja prijenosom znanja i tradicionalnog učenja su označene zvjezdicom (*).

Tablica 4. Prikaz metrika vrednovanja modela.

Broj prenesenih slojeva	Srednja vrijednost			Standardna devijacija		
	F_1	MCC	ROC-AUC	F_1	MCC	ROC-AUC
1	0.4135	0.3343*	0.8166*	0.0291	0.0305	0.0158
2	0.4596	0.3867	0.8416	0.025	0.0303	0.012
3	0.5053*	0.4368*	0.8647*	0.0227	0.0272	0.0098
4	0.5175*	0.4551*	0.8756*	0.0286	0.0317	0.0101
5	0.5237*	0.4589*	0.8729*	0.0246	0.027	0.0087
tradicionalno učenje	0.4347	0.3656	0.83	0.0333	0.0327	0.0135

Iako rezultati prikazani u tablici 4. sugeriraju da za veći broj prenesenih slojeva učenje prijenosom znanja nadmašuje tradicionalno učenje, potrebno je tu tvrdnju provjeriti statističkim testom. U ovome slučaju, zadatak statističkog testa je da utvrdi postoji li razlika između dvije skupine mjerenja odnosno koliko iznosi vjerojatnost da je uočena razlika rezultat slučajnosti.

Statistički test je potrebno odabrati u skladu s podacima i informacijama koje su nam dostupne za te podatke budući da nisu svi testovi prikladni za sve situacije. Za ovaj eksperiment odabran je Wilcoxonov signed-rank test koji je neparametarski test te ne podrazumijeva normalnu razdiobu mjerenja. Budući da su svi algoritmi bili testirani na istim podjelama skupa podataka, može se reći kako su dobivena mjerenja uparena. Ako mjerenja učenja prijenosom znanja označimo sa x , a mjerenja tradicionalnog učenja sa y , tada Wilcoxon test računa razliku $d = x - y$ te se ovisno o distribuciji d može utvrditi postoji li razlika između ove dvije skupine. Ukoliko razlika ne postoji, d će biti distribuiran oko nule, a ukoliko razlika postoji d će biti distribuiran oko neke druge točke. Ako je d pozitivan, to znači da učenje prijenosom znanja ostvaruje bolje rezultate u odnosu na tradicionalno učenje, a u obratnom slučaju isto vrijedi za tradicionalno učenje.

Kao rezultat testa dobiva se takozvana p -vrijednost, a to je vrijednost koja pokazuje kolika je vjerojatnost da je opažana razlika rezultat slučajnosti. Radi toga se definira razina značajnosti za statistički test koja se označava sa α . Ukoliko je p -vrijednost manja od razine značajnosti, odbacuje se nulta hipoteza i prihvaća se alternativna hipoteza, a inače se prihvaća nulta hipoteza i odbacuje alternativna hipoteza. Odabrani test kao nultu hipotezu tvrdi da je razlika između dva uzorka distribuirana oko nule, a alternativna hipoteza je da nije centrirana oko nule. Pregledom literature utvrdio sam kako je za područje strojnog učenja veoma čest odabir koristiti $\alpha = 0.01$ te sam odlučio istu vrijednost koristiti i u svojem radu.

Statistički test je pokazao kako je model koji ima samo jedan preneseni sloj statistički signifikantno lošiji od modela koji je dobivenim tradicionalnim učenjem. Pritom se F_1 metrika

umanjila za 4.88%, MCC metrika za 8.56% te ROC-AUC za 1.61%. U ovome slučaju je došlo do takozvanog negativnog prijenosa koji rezultira time da prijenos znanja smanji prediktivnu moć modela. Ovakva vrsta prijenosa se i dalje istražuje u znanstvenoj zajednici, a zbog slabe interpretabilnosti neuronskih mreža nije moguće intuitivno shvatiti zašto dolazi do pogoršanja.

U slučaju kada su bila prenesena prva dva sloja nije pronađena statistički signifikantna razlika za ijednu metriku. To upućuje na to da se prenošenjem prva dva sloja dobiva model koji funkcionira podjednako dobro kao i model dobiven tradicionalnim učenjem.

Prilikom prijenosa znanja iz prva četiri odnosno pet slojeva dolazi do statistički značajnog poboljšanja modela prema sve tri metrike. Međutim, pritom ROC-AUC metrika pokazuje kako se najbolji rezultat ostvaruje za četiri prenesena sloja (poboljšanje od 5.49%), a F_1 i MCC metrike pokazuju kako se najbolji rezultati ostvaruju za pet prenesenih slojeva (poboljšanja od 20.47% i 25.52%, respektivno). Ovo neslaganje metrika proizlazi iz njihove same definicije te je ovisno o primjeni modela potrebno odabrati prikladnu metriku i u skladu s njom odabrati broj prenesenih slojeva. Međutim, budući da je ROC-AUC metrika više osjetljiva na nebalansiranost u klasama, bolje se voditi po F_1 i MCC metrikama koje su otpornije na nebalansiranost.

6. Zaključak

U ovom je radu istražena primjena učenja prijenosom znanja naučenog pri učenju modela na Speech Commands skupu podataka s ciljem utvrđivanja uspješnosti ovakve metode učenja u problemu klasifikacije zvuka. Kako modeli postaju dublji i složeniji, tako raste i potreba za većim skupovima podataka za učenje. Primjena učenja prijenosom znanja je općenito dobro proučena u nekim područjima (kao što je klasifikacija slika) i pokazala se kao jedan od dobrih načina za rješavanje problema manjka podataka. Proveden je eksperiment kroz koji je ispitano učenje prijenosom znanja u situaciji unutar iste domene između dva različita skupa podataka zvučnih datoteka, gdje se u izvorišnom zadatku radi o klasifikaciji izgovorene riječi dok je u odredišnom zadatku riječ o klasifikaciji emocije kojom je izgovorena rečenica. Skup podataka u izvornom zadatku je dovoljne veličine za uspješno učenje.

Konačno, promatran je učinak učenja prijenosom znanja na performanse modela ovisno o količini prenesenog znanja te su evaluirane performanse svih modela. Eksperiment je pokazao da postoji statistički značajna razlika u točnosti između modela koji su naučeni tehnikom prijenosa znanja i modela obučenih od nule. ROC-AUC metrika je ostvarila najveće poboljšanje (5.49%) kada je preneseno znanje sa četiri sloja, dok su F_1 i MCC metrike pokazale kako se najbolji rezultati ostvaruju za pet prenesenih slojeva (poboljšanja od 20.47% i 25.52%, respektivno). Osim poboljšanja performansi, uočena je i situacija gdje je učenje prijenosom znanja rezultiralo statistički značajno lošijom izvedbom modela od modela učenog od nule (tj. negativan prijenos). U ovoj je situaciji prenesen samo jedan sloj te se pritom F_1 metrika umanjila za 4.88%, MCC metrika za 8.56% te ROC-AUC za 1.61%.

U ovome radu je došlo do pojave negativnog prijenosa što predstavlja slabo istraženu pojavu do koje može doći prilikom prijenosa znanja, a odnosi se na pogoršanje modela. Iako je moguće nagađati zašto dolazi do toga, za bolje razumijevanje ove pojave zasigurno su potrebna daljnja istraživanja.

Rezultati dobiveni u ovome radu ukazuju na to da primjena učenja prijenosom znanja za zadatak klasifikacije zvuka može rezultirati dobrim rezultatima modela ako se prenese dovoljna količina znanja. Primjena učenja prijenosom znanja za zadatak klasifikacije zvuka tek treba postati popularna i treba se gledati kao dobar izbor budući da je već revolucionirala mnoge druge domene primjene.

Reference

- [1] K. Weiss, T. Khoshgoftaar i D. Wang, »A survey of transfer learning,« *J Big Data*, svez. 3, br. 9, 2016.
- [2] L. Torrey i J. Shavlik, *Transfer Learning*, Madison: IGI Global, 2010.
- [3] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong i Q. He, »A Comprehensive Survey on Transfer Learning,« *Proceedings of the IEEE*, svez. 109, br. 1, pp. 43-76, 2021.
- [4] S. J. Pan i Q. Yang, »A Survey on Transfer Learning,« *IEEE Transactions on Knowledge and Data Engineering*, svez. 22, br. 10, pp. 1345-1359, 2010.
- [5] K. Palanisamy, D. Singhanian i A. Yao, »Rethinking CNN Models for Audio Classification,« *Computer Vision and Pattern Recognition*, 2020.
- [6] S. Jadon, »Introduction to Different Activation Functions for Deep Learning,« 2018. [Mrežno]. Available: <https://medium.com/@shrutijadon/survey-on-activation-functions-for-deep-learning-9689331ba092>. [Pokušaj pristupa 16. studeni 2022].
- [7] H. Fan, M. Jiang, L. Xu, H. Zhu, J. Cheng i J. Jiang, »Comparison of Long Short Term Memory Networks and the Hydrological Model in Runoff Simulation,« *Water*, svez. 12, br. 1, 2020.
- [8] D. Berrar, *Cross-Validation*, 2019.
- [9] X. Ying, »An overview of overfitting and its solutions,« *Journal of physics: Conference series*, svez. 1168, br. 2, 2019.
- [10] P. Warden, »Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition,« *Computation and Language*, 2018.
- [11] E. Otović, M. Njirjak, D. Jozinović, G. Mauša, A. Michelini i I. Štajduhar, »Intra-domain and cross-domain transfer learning for time series data – How transferable are the features?,« *Knowledge-Based System*, svez. 239, p. 107976, 2022.
- [12] S. R. Livingstone i F. A. Russo, »The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English,« *PLoS ONE*, svez. 13, br. 5, 2018.
- [13] D. P. Kingma i J. Ba, »Adam: A Method for Stochastic Optimization,« u *3rd International Conference for Learning Representations*, San Diego, 2015.
- [14] M. D. Zeiler, »ADADELTA: An Adaptive Learning Rate Method,« *Machine Learning*, 2012.
- [15] T. Kurbiel i S. Khaleghian, »Training of Deep Neural Networks based on Distance Measures using RMSProp,« *Machine Learning*, 2017.

Popis slika

Slika 1. Model neurona s tri ulaza.....	4
Slika 2. Povratni LSTM stroj [7].....	6
Slika 3. 10-struka unakrsna provjera valjanosti.	13
Slika 4. Prikaz slojeva modela.	16
Slika 5. Proces učenja prijenosom znanja.	19

Popis tablica

Tablica 1. Pregled najkorištenijih aktivacijskih funkcija [6].	5
Tablica 2. Matrica zabune.	8
Tablica 3. Usporedba skupova podataka.	14
Tablica 4. Prikaz metrika vrednovanja modela.	20

Popis priloga

- A. scripts/preprocess_datasets.sh – Skripta za pokrenuti pretprocesiranje oba skupa podataka.
- B. scripts/train_models.sh – Skripta za pokrenuti tradicionalno učenje nad oba skupa podataka.
- C. scripts/train_tl_models.sh – Skripta za pokrenuti učenje prijenosom znanja iz modela naučenog na Speech Commands skupu podataka u novi model koji za učenje koristi RAVDESS skup podataka. Pritom se uči više modela gdje se prenosi različita količina znanja.
- D. scripts/evaluate_models.sh – Skripta za pokrenuti evaluacije tradicionalno učenih modela.
- E. scripts/evaluate_tl_models.sh – Skripta za pokrenuti evaluacije modela učenih tehnikom prijenosa znanja.
- F. scripts/visualize_models.sh – Skripta za generirati vizualizacije tradicionalno učenih modela.
- G. scripts/visualize_tl_models.sh – Skripta za generirati vizualizacije modela učenih tehnikom prijenosa znanja.
- H. preprocess_ravdess.py – Python datoteka koja sadrži programski kôd za pretprocesiranje RAVDESS skupa podataka.
- I. preprocess_speech.py – Python datoteka koja sadrži programski kôd za pretprocesiranje Speech Commands skupa podataka.
- J. Models.py – Python datoteka koja sadrži programski kôd za definiciju modela.
- K. train.py – Python datoteka koja sadrži programski kôd za tradicionalno učenje ili učenje prijenosom znanja.
- L. evaluate.py – Python datoteka koja sadrži programski kôd za evaluaciju modela.
- M. visualize.py – Python datoteka koja sadrži programski kôd za vizualizaciju modela.