



UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET



Ivana Čavor

**Procjena brzine vozila na osnovu zvučnog
signala primjenom konvolucionih neuralnih
mreža**

– magistarski rad –

Podgorica, septembar 2023.

UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET

Ivana Čavor

**Procjena brzine vozila na osnovu zvučnog
signala primjenom konvolucionih neuralnih
mreža**

– magistarski rad –

Podgorica, septembar 2023.

PODACI I INFORMACIJE O MAGISTRANTU

Ime i prezime: **Ivana Čavor**

Datum i mjesto rođenja: 29.08.1993

Prethodno završene studije:

Elektrotehnički fakultet, osnovne akademske studije (180 ECTS kredita),
studijski program: Elektronika, telekomunikacije i računari, 2015.

Elektrotehnički fakultet, specijalističke akademske studije (60 ECTS kredita),
studijski program: Elektronika, telekomunikacije i računari, smjer Računari,
2016.

INFORMACIJE O MAGISTARSKOM RADU

Elektrotehnički fakultet

Studijski program: Elektronika, telekomunikacije i računari

Naslov rada: **Procjena brzine vozila na osnovu zvučnog signala primjenom
konvolucionih neuralnih mreža**

Mentor: Prof. dr Slobodan Đukanović

UDK, OCJENA I ODBRANA MAGISTARSKOG RADA

Datum prijave magistarskog rada: 5.12.2022.godine

Datum sjednice Vijeća na kojoj je prihvaćena tema: 23.3.2023. godine

Komisija za ocjenu teme i podobnosti magistranta:

1. Prof. dr Miloš Daković
2. Prof. dr Slobodan Đukanović
3. Prof. dr Vesna Popović - Bugarin

Komisija za ocjenu rada:

1. Prof. dr Miloš Daković
2. Prof. dr Slobodan Đukanović
3. Prof. dr Vesna Popović - Bugarin

Komisija za odbranu rada:

1. Prof. dr Miloš Daković
2. Prof. dr Slobodan Đukanović
3. Prof. dr Vesna Popović - Bugarin

Datum odbrane: 15.9.2023.godine

Datum promocije: _____

Ime i prezime autora: Ivana Čavor, Spec. Sci

E T I Č K A I Z J A V A

U skladu sa članom 22 Zakona o akademskom integritetu i članom 24 Pravila studiranja na postdiplomskim studijama, pod krivičnom i materijalnom odgovornošću, izjavljujem da je magistarski rad pod naslovom

**„ PROCJENA BRZINE VOZILA NA OSNOVU ZVUČNOG SIGNALA
PRIMJENOM KONVOLUCIONIH NEURALNIH MREŽA “**

moje originalno djelo.

Podnosilac izjave,

Ivana Čavor, Spec. Sci



U Podgorici, dana 3.7.2023. godine

Sadržaj

Sažetak	1
Abstract	2
1 Uvod	3
2 Konvolucione neuralne mreže	8
2.1 Vještačke neuralne mreže	9
2.1.1 Aktivacione funkcije	11
2.1.2 Funkcija gubitka	18
2.1.3 Procjena uspješnosti modela	20
2.2 Duboko učenje	23
2.2.1 Propagacija unaprijed i unazad	24
2.2.2 Regularizacija	28
2.2.3 Optimizacioni algoritmi	32
2.3 Istorija i razvoj konvolucionih neuralnih mreža	33
2.4 Karakteristike konvolucionih neuralnih mreža	36
2.4.1 Arhitektura konvolucione neuralne mreže	38
3 Procjena brzine i klase brzine vozila	44
3.1 Unakrsna validacija	46
3.2 Trening procedure	47
3.2.1 Pristup sa optimalnom epohom	47
3.2.2 Pristup sa minimalnim gubitkom na validacionom skupu	49
4 Eksperimentalni rezultati	50
4.1 Skup podataka	50
4.1.1 Prikupljanje podataka	51
4.1.2 Anotacija podataka	54
4.2 Prikaz rezultata	54

4.2.1	Procjena brzine kao regresioni problem	54
4.2.2	Procjena brzine kao klasifikacioni problem	56
5	Zaključak	59

Sažetak

Precizna i pouzdana procjena brzine vozila je od ključne važnosti za unapređenje kvaliteta saobraćajnih usluga i efikasnu primjenu zakonskih regulativa sa ciljem poboljšanja bezbjednosti u saobraćaju. Dok su najnoviji sistemi za nadgledanje saobraćaja uglavnom bazirani na kompjuterskoj viziji, zvučni sistemi za nadgledanje saobraćaja zasnovani na vještačkoj inteligenciji su još u inicijalnoj fazi razvoja. U skladu s tim, ovaj magistarski rad ima za cilj da istraži mogućnost procjene brzine vozila na osnovu zvuka koji vozilo proizvodi prilikom prolaska pored zvučnog senzora (mikrofona).

U poređenju sa drugim oblastima istraživanja, broj dostupnih skupova podataka za procjenu brzine vozila je vrlo ograničen. Stoga je za potrebe ovog istraživanja formiran anotirani skup zvučnih zapisa kretanja raznovrsnih vozila, u stvarnim uslovima, koji se može koristiti za procjenu brzine vozila. Procjena brzine je posmatrana kao regresioni i klasifikacioni problem, pri čemu se koriste duboke jednodimenzionalne konvolucione mreže koje su prilagođene analizi i obradi vremenske reprezentacije zvučnih signala. Glavna prednost konvolucionih neuralnih mreža ogleda se u njihovoj sposobnosti da automatski identifikuju ključne karakteristike zvučnog signala vozila i na taj način omoguće efikasnu i objektivnu procjenu brzine vozila.

Rezultati istraživanja pokazuju da se jednodimenzionalne konvolucione neuralne mreže mogu uspješno primijeniti za procjenu brzine vozila na osnovu zvuka. Takođe, rad pruža dublje razumijevanje uticaja različitih faktora na kvalitet procjene brzine vozila i daje smjernice za moguća rješenja za njihovo prevazilaženje, koja će biti predmet budućih istraživanja.

Ključne riječi: procjena brzine vozila, duboko učenje, jednodimenzionalne konvolucione neuralne mreže

Abstract

Accurate and reliable vehicle speed estimation is crucial for improving the quality of transportation services and the effective implementation of traffic regulations aimed at enhancing road safety. While the latest traffic monitoring systems are predominantly based on computer vision, sound-based traffic monitoring systems that utilize artificial intelligence are still in the early stages of development. Accordingly, this master's thesis aims to explore the possibility of estimating vehicle speed based on the sound produced by the vehicle when passing by a acoustic sensor (microphone).

Compared to other research areas, the number of available datasets for vehicle speed estimation is very limited. Therefore, for the purposes of this research, an annotated dataset of sound recordings of diverse vehicles in real-world conditions was created, which can be used for vehicle speed estimation. Speed estimation is treated as a regression and classification problem, utilizing deep one-dimensional convolutional neural networks that are tailored for the analysis and processing of the temporal representation of sound signals. The main advantage of convolutional neural networks lies in their ability to automatically identify key characteristics of the vehicle sound signal, enabling efficient and objective vehicle speed estimation.

The research results demonstrate that one-dimensional convolutional neural networks can be successfully applied to estimate vehicle speed based on sound. Additionally, the study provides a deeper understanding of the impact of different factors on the quality of vehicle speed estimation and offers guidelines for potential solutions to overcome them, which will be the subject of future research.

Keywords: vehicle speed estimation, deep learning, one-dimensional convolutional neural network

Glava 1

Uvod

Rastući trend urbanizacije za posljedicu ima povećanje broja vozila na cestama, što predstavlja veliki izazov u pogledu praćenja i kontrole saobraćaja. Prekoračenje brzine je jedan od najčešćih uzroka saobraćajnih nesreća koje mogu imati fatalne posljedice po živote ljudi i materijalnu imovinu. Stoga, efikasna procjena brzine vozila na putevima može smanjiti rizik i povećati sigurnost u saobraćaju. Primjena naprednih tehnologija baziranih na vještačkoj inteligenciji (eng. *Artificial Intelligence* - AI) u nadgledanju saobraćaja postaje sve popularnija, jer omogućava praćenje vozila na putevima, procjenu brzine, prepoznavanje registarskih tablica, procjenu brzine i detekciju prekoračenja brzine, kao i detekciju drugih saobraćajnih prekršaja.

Inteligentni transportni sistemi (eng. *Intelligent Transportation Systems* - ITS), u sklopu koncepta pametnih gradova, široko se primjenjuju u većini razvijenih zemalja radi praćenja i poboljšanja sigurnosti saobraćaja. Različiti moderni senzori omogućavaju prikupljanje raznovrsnih podataka (tekst, zvuk, slika, video) o funkcionisanju saobraćaja i njihovu obradu u stvarnom vremenu [1]. Informacije o broju vozila na putevima, procijenjenoj brzini i ubrzanju vozila, tipu vozila, kao i o eventualnim saobraćajnim nesrećama mogu poboljšati kvalitet saobraćajnih usluga i pomoći u sprovođenju zakonskih regulativa radi povećanja bezbjednosti saobraćaja na viši nivo. Wilson i kolege navode da je broj prekoračenja brzine i sudara smanjen za 35% i 25% respektivno.

Najnoviji sistemi za nadgledanje saobraćaja uglavnom su bazirani na kompjuterskoj viziji (eng. *Computer Vision* - CV) i omogućavaju detekciju, praćenje i klasifikaciju vozila na osnovu digitalnih slika. Autori rada [3] su napravili sveobuhvatan pregled najsavremenijih tehnika za vizuelni nadzor saobraćaja iz perspektive kompjuterske vizije. Rad je

fokusiran na različite pristupe i tehnologije za detekciju, prepoznavanje i praćenje vozila na osnovu slika. Autori su predstavili veliki broj tehnika, počevši od jednostavnog poređenja uzastopnih slika do izdvajanja bitnih karakteristika kao što su boja, tekstura, oblik i obrasce kretanja vozila kako bi se razvio probabilistički model koji se može koristiti za detekciju i klasifikaciju vozila. Autori su istakli da postoji potreba za daljim istraživanjem i razvojem CV tehnologija posebno u urbanim sredinama. Naime, u gradskim sredinama postoji veća gustina saobraćaja, mogu se javiti situacije u kojima vozila većih dimenzija zaklanjaju manja vozila, pozicija i orijentacija vozila na raskrsnicama se ne podudaraju sa pozicijom kamera, što značajno utiče na performanse vizuelnih nadzornih sistema. Pored toga, tehnike bazirane na slikama nailaze na kritike zbog potencijalnog kršenja privatnosti putnika u vozilu i otkrivanja njihovog identiteta. Neki od najvažnijih izazova za procjenu brzine vozila na osnovu slika su istaknuti u radu [4]. Autori su predstavili najnovije pristupe za određivanje udaljenosti i brzine vozila na osnovu slika, kao i dostupne skupove podataka i najčešće korištene metrike za određivanje greške pri procjeni brzine. Takođe, navedene su i glavne prepreke za široku primjenu kamera za praćenje saobraćaja. Nedostatak resursa i izazovi koji se javljaju u stvarnim uslovima na putevima, poput promjene osvetljenja, različitih vremenskih uslova i gužvi na putevima, predstavljaju osnovna ograničenja u vizuelnim nadzornim sistemima. Autori su takođe zaključili da je broj dostupnih skupova podataka za procjenu brzine vozila vrlo ograničen, uglavnom zbog visoke kompleksnosti i troškova procesa snimanja i označavanja podataka. Stoga, jedan od istraživačkih ciljeva ovog magistarskog rada je stvaranje anotiranog skupa audio i video zapisa kretanja različitih vozila na putevima, u stvarnim uslovima, koji se može koristiti za procjenu brzine vozila [5].

Autori rada [6] smatraju da je upotreba i analiza velike količine podataka (eng. *big data*) sa saobraćajnih puteva ključna za budući razvoj ITS tehnologija. Tradicionalne metode mašinskog učenja (eng. *Machine Learning* - ML) u velikoj mjeri zavise od ekspertskog znanja ML inženjera i ne postižu dobre rezultate u slučaju heterogenih ulaznih podataka. S druge strane, korišćenje dubokog učenja (eng. *Deep Learning* - DL) u kombinaciji sa velikim količinama heterogenih podataka prikupljenih sa saobraćajnih puteva donosi značajne prednosti u ITS oblasti. DL metode su razvijene po ugledu na strukturu i sistem obrade informacija koje koristi ljudski mozak. Ljudski mozak ima sposobnost da automatski prepoznaje i izdvaja relevantne informacije iz okoline koja nas okružuje. Na primjer, kada posmatramo sliku ili video, naš mozak može prepoznati objekte, lica, boje, pokrete i druge karakteristike u tim prikazima. Ovo se događa gotovo automatski, bez potrebe

da se posebno fokusiramo na svaki detalj. Duboko učenje se oslanja na slične principe prepoznavanja i izdvajanja informacija iz podataka, kao što to radi ljudski mozak.

U osnovi dubokog učenja su duboke neuralne mreže (eng. *Deep Neural Networks* - DNN) koje posjeduju sposobnost automatskog prepoznavanja informacija u složenim podacima poput zvuka, slika i video zapisa. One mogu izvući važne informacije bez potrebe za ručnim odabirom karakteristika ulaznih podataka i ekspertskim znanjem u određenoj oblasti primjene. Posebno popularan pristup u dubokom učenju su konvolucione neuralne mreže (eng. *Convolutional Neural Networks* - CNN) [7]. Dok su dvodimenzionalne CNN mreže (2D CNN) pronašle masovnu upotrebu u zadacima kompjuterske vizije, jednodimenzionalne CNN mreže (1D CNN) su primarno namijenjene za obradu jednodimenzionalnih sekvenci, kakve su zvučni signali.

U suštini, iako imaju dobre performanse, sistemi bazirani na kompjuterskoj viziji su skupi, komplikovani za instalaciju i održavanje, pritom i memorijski zahtjevni sa stanovišta čuvanja i obrade podataka. Akustični senzori zahtijevaju znatno manje resursa te se nameću kao dobra alternativa vizuelnim nadzornim sistemima. Pored toga, senzori zvuka su neosjetljivi na razne vizuelne prepreke i ometanja poput varijacija osvjetljenja. Takođe, značajna prednost akustičnih sistema za nadgledanje saobraćaja, u kontekstu procjene brzine, je mogućnost procjene brzine i prije i poslije pozicije senzora. Naime, akustični senzori mogu registrovati zvuk i nakon što vozilo prođe pored mikrofona, što nije moguće kod vizualnog nadgledanja saobraćaja zasnovanog na analizi digitalne slike. Akustični sistemi za nadgledanje saobraćaja koji se oslanjaju na vještačku inteligenciju još uvijek su u početnoj fazi razvoja u odnosu na vizuelne nadzorne sisteme, što ukazuje na potrebu za daljim istraživanjima u ovoj oblasti. Stoga, jedan od ciljeva ovog magistarskog rada je da doprinese razvoju i primjeni vještačke inteligencije u akustičnom nadgledanju saobraćaja.

Fizičke osobine zvuka poput pritiska, frekvencije i intenziteta, osnovne su karakteristike koje utiču na ljudsku percepciju zvuka. Na primjer, na osnovu promjena u intenzitetu zvuka koji proizvodi vozilo, čovjek može procijeniti da li se ono kreće brže ili sporije, iz kog pravca dolazi, da li se udaljava ili približava. Takođe, većina vozača ima sposobnost da prepozna da nešto nije u redu sa njihovim vozilom na osnovu promjene karakterističnog zvuka vozila. Ovaj rad ima za cilj da pokaže da se brzina vozila može precizno i objektivno procijeniti iz jednokanalnog zvučnog signala korištenjem 1D CNN mreža, koje su prilagođene analizi i obradi vremenske reprezentacije zvučnih signala. Njihova struktura omogućava efikasnu obradu podataka i brze računске performanse, što ih čini pogodnim alatom procjenu brzine vozila u stvarnim saobraćajnim uslovima.

Postojeći pristupi koji se bave procjenom brzine vozila na osnovu zvuka mogu se podijeliti na sisteme bazirane na jednom mikrofону i one koje pri akviziciji koriste više mikrofona. Međutim, složenost obrade višekanalnih zvučnih signala i komplikovana instalacija akustičnog sistema sa više mikrofona usmjerila je interesovanje za procjenu brzine vozila na model jednokanalnog zvučnog signala. Autori rada [8] predlažu upotrebu pasivnog senzora za procjenu brzine vozila na osnovu Doplerovog efekta. Procjena brzine i položaja vozila dobijena je analizom karakterističnih promjena u spektrogramu zvučnog signala vozila. Utvrđeno je da se Doplerovi pomjeraji u zvuku, koji se odnose na brzinu, pojavljuju kao „šavovi“ (eng. *seams*). Prikazani rezultati pokazuju da se procijenjene vrijednosti brzine vrlo dobro podudaraju sa stvarnim vrijednostima. U radu [9] autori koriste spektralne i prostorne karakteristike akustičnih podataka i kombinuju više različitih algoritama mašinskog učenja da bi poboljšali preciznost procjene brzine. Isti navode da u sistemima sa jednim mikrofonom, informacije o frekvenciji signala zasnovane na Doplerovom efektu nisu korisne za procjenu brzine vozila. Promjena brzine vozila, u smislu detekcije ubrzanja, usporenja i održavanja konstantne brzine, detaljno je obrađena u [10]. Autori su uzeli u obzir različite reprezentacije audio signala vozila i kreirali 85 diskriminativnih karakteristika na osnovu kojih su primjenom različitih ML metoda (metod potpornih vektora, *random forest* metode i neuralne mreže) vršili detekciju i klasifikaciju promjene brzine. Za procjenu brzine vozila u [11], vremensko-frekvencijske karakteristike zvučnog signala koji proizvodi vozilo koriste se kao ulaz u neuralnu mrežu. U istraživanju koje je sprovedeno, korištenjem neuralne mreže sa dva skrivena sloja, postignuta je prosječna greška procjene brzine od 14.57% za brzine vozila između 10 i 40 km/h. Autori navode da se predloženi pristup može koristiti u gradskim područjima kao komponenta sistema upozorenja na pješačkim prelazima, gdje se nesreće najčešće dešavaju zbog nepažnje pješaka prilikom prelaska ulice.

Đukanović i ostali, u radu [12], uvode novu analitičku funkciju zavisnu od brzine, pod nazivom modifikovano slabljenje (eng. *Modified Attenuation-MA*), po uzoru na slabljenje amplitude zvučnog signala u odnosu na rastojanje do izvora zvuka. Predložena MA karakteristika procjenjuje se na osnovu log-mel spektrograma zvučnog signala vozila koristeći duboke neuralne mreže. Na osnovu ove karakteristike, procijenjena je brzina vozila koristeći regresioni metod potpornih vektora (eng. *support vector regression*). MA karakteristika primijenjena je u [13] za korekciju tačnog trenutka prolaska vozila pored zvučnog senzora, što je dovelo do smanjenja srednje greške procjene brzine vozila sa 7.39 km/h na 6.92 km/h. Mel-frekvencijske reprezentacije (mel-spektrogram, log-mel

spektrogram, log-mel cepstralni spektrogram) zvučnog signala vozila i potpuno povezane neuralne mreže su razmatrane u [14]. Ekspertimentalni rezultati pokazuju da pažljivo odabrane mel-frekvencijske karakteristike mogu da se koriste direktno u procjeni brzine, bez potrebe za drugim karakteristikama kreiranim od strane eksperta.

Treba pomenuti da svi prethodno navedeni pristupi koriste neku vrstu obrade zvučnog signala prije procjene brzine vozila. Glavna prednost konvolucionih neuralnih mreža, u odnosu na postojeće metode za procjenu brzine vozila, ogleda se u mogućnosti automatskog lociranja i selektovanja karakteristika iz vremenske reprezentacije zvučnog signala koje su relevantne za procjenu brzine vozila. Na ovaj način se eliminišu subjektivne procjene i greške koje mogu nastati pri ručnom izboru karakteristika. Uz to, upotreba CNN mreža donosi i druge prednosti u odnosu na tradicionalne metode. CNN mreže imaju visoke računске performanse i mogu efikasno obraditi velike količine podataka u stvarnom vremenu. Ovo je od suštinske važnosti u saobraćajnim okruženjima gdje brza i tačna procjena brzine vozila može biti ključna za sigurnost i upravljanje saobraćajem.

Magistarski rad se sastoji od pet poglavlja. U prvom poglavlju magistarskog rada data su uvodna razmatranja vezana za procjenu brzine vozila na osnovu zvuka, predmet i motivi istraživanja. Takođe, dat je pregled dosadašnjih istraživanja u oblasti procjene brzine vozila na osnovu zvučnog signala koje proizvodi vozilo tokom prolaska pored zvučnog senzora. U drugom poglavlju detaljno su opisane konvolucione mreže, njihov princip rada i područja njihove primjene, sa posebnim osvrtom na jednodimenzionalne konvolucione mreže. Predloženi 1D CNN model je predstavljen u trećem poglavlju magistarskog rada. Takođe, opisan je proces treniranja i validacije 1D CNN modela za potrebe procjene brzine vozila na osnovu zvučnog signala. Četvrto poglavlje predstavlja korišćeni skup anotiranih zvučnih zapisa, kao i rezultate procjene brzine na datom skupu. Procjena brzine vozila se posmatra kao regresioni problem, kad se procjenjuje tačna vrijednost brzine, i kao klasifikacioni problem, kad se procjenjuje kojoj klasi (interval brzine određene širine) pripada data brzina. U okviru ovog poglavlja su prikazani rezultati oba korišćena pristupa. Zaključna razmatranja i mogući pravci daljeg istraživanja u oblasti procjene brzine vozila na osnovu zvučnog signala prikazani su u petom poglavlju. Na poslednjim stranicama rada je priložen spisak literature koja je korišćena za izradu magistarskog rada.

Glava 2

Konvolucione neuralne mreže

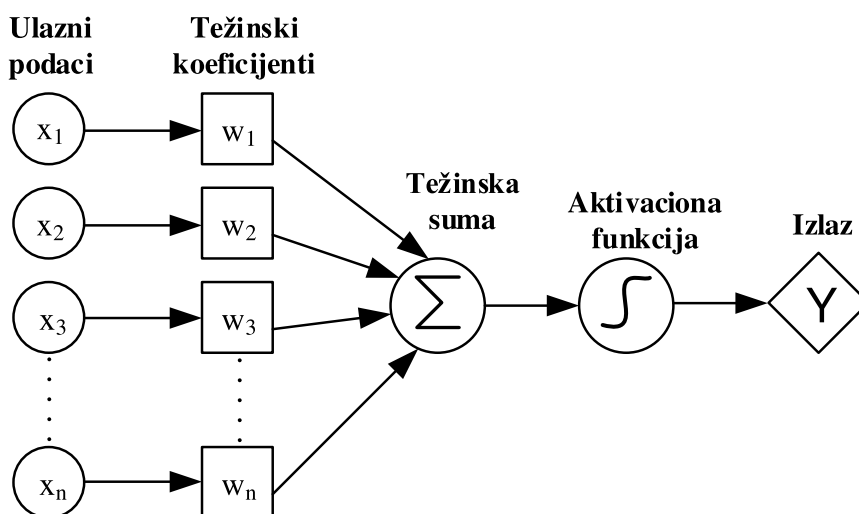
Vještačka inteligencija je doživjela značajan napredak u mnogim područjima u posljednjih nekoliko godina. Posebno su impresivni rezultati postignuti u oblasti kompjuterske vizije, što je primatno omogućeno razvojem konvolucionih neuralnih mreža. U dubokom učenju, konvolucione neuralne mreže predstavljaju posebnu vrstu vještačkih neuralnih mreža koje su prilagođene za obradu signala sa prostornom i vremenskom strukturom. Duboke dvodimenzionalne konvolucione neuralne mreže su veoma efikasne u analizi vizualnih podataka jer mogu automatski identifikovati bitne karakteristike i obrasce u slikama i video zapisima, čime simuliraju rad ljudskog mozga u obradi vizuelnih informacija. Postojanje velikih skupova označenih podataka je omogućilo 2D CNN mrežama da nauče složene obrasce i objekte na slikama i da postignu visoku preciznost u klasifikaciji, prepoznavanju i segmentaciji slika. Potreba za efikasnom obradom sekvencijalnih podataka u različitim oblastima kao što su analiza zvuka, biomedicinski signali i senzorski podaci, postavila je izazove pred istraživače. U tom kontekstu, modeli 1D CNN mreža su se pokazali kao ključni faktor u ostvarivanju napretka. Njihova sposobnost efikasnog učenja iz sekvencijalnih podataka omogućila je razvoj inovativnih rješenja, otvorila nove mogućnosti za analizu i obradu sekvencijalnih informacija i unaprijedila performanse u tim oblastima. Glavna prednost jednodimenzionalnih konvolucionih neuralnih mreža leži u njihovoj jednostavnoj i kompaktnoj konfiguraciji, što omogućava jeftinu i izvodljivu implementaciju u stvarnim situacijama [15].

Ovaj rad ima za cilj da pokaže da se brzina vozila može procijeniti iz jednokanalnog zvučnog signala primjenom 1D CNN mreža. Stoga će ovo poglavlje obuhvatiti detaljan opis principa rada konvolucionih neuralnih mreža, uključujući operaciju konvolucije i

različite vrste slojeva koje se koriste u ovim mrežama. Prvenstveno će biti detaljno opisane vještačke neuralne mreže (eng. *Artificial Neural Networks* - ANN), kao i ključni koncepti koji su neophodni za razumijevanje funkcionisanja bilo koje neuralne mreže uključujući i konvolucione neuralne mreže.

2.1 Vještačke neuralne mreže

Vještačke neuralne mreže su matematički modeli inspirisani strukturom i funkcijom bioloških neurona u čovjekovom nervnom sistemu. Glavna ideja ovih modela je da simuliraju biološke procese razmjene informacija u mozgu kako bi se postiglo inteligentno ponašanje računarskog sistema. Naime, u ljudskom mozgu, neuroni su povezani sa hiljadama drugih neurona u složenoj mreži koja omogućava prenos informacija i obradu signala. Svaki neuron prima signale iz mnogih drugih neurona, obrađuje te signale, a zatim šalje izlazni signal drugim neuronima. Ova složena mreža veza između neurona omogućava mozgu da obavlja složene funkcije poput obrade senzornih informacija, kretanja, emocija i kognitivnih procesa. Na sličnom principu se vrši razmjena informacija između neuronskih slojeva u ANN mreži. Uzmimo za primjer perceptron, najjednostavniji model neuralne mreže koji je predložio Frank Rosenblatt u svom radu iz 1958. godine [16]. Rosenblatt perceptron, prikazan na slici 2.1, sastoji se od jednog neurona koji prima više ulaznih signala, obrađuje ih i daje jedan izlazni rezultat, koji se koristi za klasifikaciju ulaznih uzoraka.



Slika 2.1: Perceptron

Ulazni podaci perceptrona su karakteristike koje opisuju neki objekat ili pojavu i obično se predstavljaju kao vektori koji se sastoje od numeričkih vrijednosti. Na slici 2.1, vrijednosti karakteristika: x_1, x_2, \dots, x_{n-1} i x_n koriste se kao ulazni podaci perceptrona. Često se ulazni podaci smiještaju u takozvani ulazni sloj, koji ima jedan neuron za svaki ulazni atribut ili karakteristiku. Međutim, obzirom da ti neuroni ne obrađuju već čuvaju podatke, ulazni sloj je samo prolazni sloj koji usmjerava podatke dalje u mrežu.

Može se reći da važnost određene ulazne karakteristike u procesu donošenja odluke o pripadnosti ulaznog uzorka određenoj klasi zavisi od težinskih koeficijenata neurona. Težinski koeficijenti: w_1, w_2, \dots, w_{n-1} i w_n , sa slike 2.1, predstavljaju parametre modela koje mreža uči kroz proces treniranja kako bi klasifikovala nove uzorke u ispravne klase. Vrijednost težinskih koeficijenata direktno utiče na doprinos ulaznog signala ukupnom izlaznom signalu. Naime, svaki podatak iz ulaznog sloja se množi sa odgovarajućim težinskim koeficijentom i kreira se takozvana težinska suma koja je jednaka:

$$u = w_1x_1 + w_2x_2 + \dots + w_nx_n. \quad (2.1)$$

Ukoliko vektor ulaznih karakteristika označimo sa $X = [x_1, x_2, \dots, x_n]^T$ i vektor težinskih koeficijenata sa $W = [w_1, w_2, \dots, w_n]$, prethodno definisanu težinsku sumu možemo zapisati kao:

$$W^T \cdot X = \sum_{i=1}^n w_i x_i. \quad (2.2)$$

Često se na težinsku sumu dodaje parametar b , koji se definiše kao sistemska greška (eng. *bias*) i omogućava da model nauči neke složenije oblike odlučivanja. Naime, *bias* utiče na prag aktivacije neurona, te će postavljanje visoke vrijednosti za *bias* povećati vjerovatnoću aktivacije neurona, dok će postavljanje niske vrijednosti za *bias* smanjiti tu vjerovatnoću. Dakle, kada se uzme u obzir *bias*, jednačina 2.1 se može zapisati kao:

$$u = W^T \cdot X + b. \quad (2.3)$$

Nakon što se izračuna težinska suma ulaza u neuralnu mrežu, ona se transformiše u izlazni signal \hat{y} primjenom odgovarajuće aktivacione funkcije (eng. *activation function*) f , odnosno:

$$\hat{y} = f(W^T \cdot X + b). \quad (2.4)$$

Jedan perceptron može klasifikovati ulazne podatke koji su linearno odvojivi, što znači da se podaci mogu jasno razdvojiti u dvije klase. Dakle, izlazni signal perceptrona uzima vrijednost 0 ili 1, što se često tumači kao negativna ili pozitivna klasifikacija. Perceptron donosi odluku o tome da li ulazni podatak pripada određenoj klasi ili ne, na osnovu naučenih težina i praga (eng. *threshold*) koji određuju granicu za donošenje odluke.

Uobičajeni postupak je da se početni težinski koeficijenti nasumično generišu, a zatim se postepeno podešavaju tokom treninga. Optimizacija parametara modela tokom procesa učenja vrši se na način da se minimizuje funkcija gubitka (eng. *loss function*), koja predstavlja mjeru razlike između stvarnih vrijednosti i predviđenih vrijednosti izlaza. Dakle, cilj procesa obuke perceptrona i drugih neuralnih mreža je naučiti optimalne težinske koeficijente koji će omogućiti modelu da minimizuje funkciju gubitka i da ima dobre performanse u predviđanju izlaza za nove, nepoznate uzorke. Postoje razni algoritmi koji se koriste za pronalaženje optimalnih vrijednosti težinskih koeficijenata u procesu učenja neuralne mreže, odnosno za minimizaciju funkcije gubitka i o njima će kasnije biti riječi.

Aktivaciona funkcija, funkcija gubitka i optimizacioni algoritmi su tri ključna elementa u dizajniranju neuralne mreže i od velike su važnosti za postizanje visokih performansi u rješavanju različitih vrsta problema. Stoga, u sljedećim podsekcijama biće više riječi o najčešće korišćenim aktivacionim funkcijama, funkcijama gubitka i algoritmima optimizacije.

2.1.1 Aktivacione funkcije

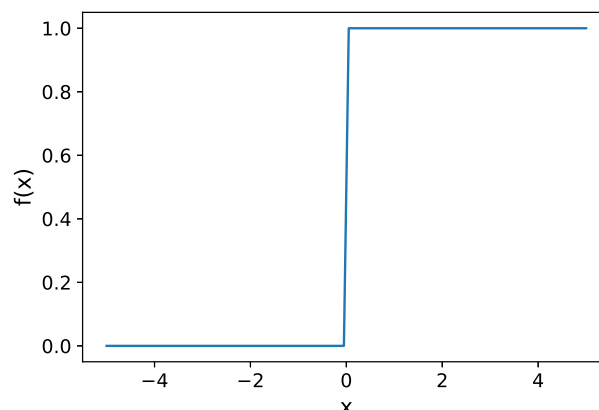
Aktivaciona funkcija neuralne mreže je matematička funkcija koja se primjenjuje na izlazu svakog neurona u mreži. Uloga aktivacione funkcije je vrlo važna u procesu odlučivanja o aktivaciji neurona, a samim tim i u konačnom izlazu neuralne mreže. Ova funkcija uzima težinsku sumu ulaznih karakteristika i težina neurona kao ulaz i daje izlazni signal, koji se zatim koristi za prenošenje informacija kroz neuralnu mrežu. Ukoliko se izlaz neurona računa kao linearna transformacija ulaza, kao u jednačini 2.3, tada govorimo o linearnoj aktivacionoj funkciji. Neuralna mreža sa linearnom aktivacionom funkcijom, bez obzira koliko slojeva da ima, nije u mogućnosti da nauči kompleksne obrasce u podacima, jer će se svi slojevi ponašati na isti način, odnosno vršiće linearnu transformaciju ulaza. Obzirom da realni podaci pretežno ispoljavaju nelinearnu zavisnost, u višeslojnim neuralnim mrežama koriste se nelinearne aktivacione funkcije. Linearana aktivaciona funkcija se uglavnom koristi u posljednjem sloju ANN mreže kod regresionih problema, gdje se

očekuje samo jedna realna vrijednost kao rezultat.

Aktivaciona funkcija perceptrona je obično **step funkcija**, koja ima vrijednost 1 ako je ulaz veći od praga (eng. *threshold*) i vrijednost 0 u suprotnom. Matematički izaz za *step* funkciju može se predstaviti jednačinom:

$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x \geq 0. \end{cases} \quad (2.5)$$

Prethodnu relaciju možemo protumačiti na sljedeći način: ako je težinska suma ulaznih karakteristika i težina veća od praga aktivacije, perceptron će biti *uključen* i dati izlazni signal, što se tumači kao pozitivna klasa, a ako je manja od praga aktivacije, perceptron će biti *isključen* i dati izlazni signal 0, što se tumači kao negativna klasa. Dakle, *step* funkcija je binarna aktivaciona funkcija, jer ima samo dvije moguće vrijednosti za izlaz: 0 ili 1. Jasno je da *step* funkcija nije pogodna za rješavanje složenih problema koji zahtijevaju nelinearno izračunavanje zbog svoje diskretne prirode, što se može vidjeti na slici 2.2.



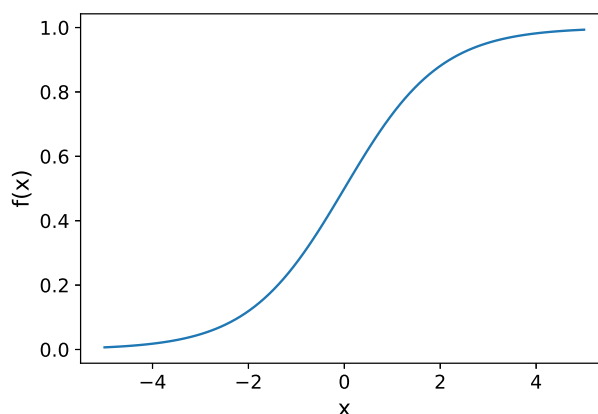
Slika 2.2: Step aktivaciona funkcija

Takođe, može se primijetiti da je *step* funkcija diskontinualna funkcija koja ima skok iz vrijednosti 0 u vrijednost 1 u tački $x = 0$. Ovo može predstavljati problem u procesu optimizacije neuralne mreže, posebno kada se vrši iterativno ažuriranje parametara mreže u smjeru negativnog gradijenta funkcije gubitka, jer se *step* funkcija ne može diferencirati (izvoditi) u tački gdje dolazi do skoka.

Umjesto *step* aktivacione funkcije, zbog prethodno navedenih ograničenja, često se koristi **sigmoidna funkcija** čiji je matematički zapis:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (2.6)$$

Sigmoidna funkcija je vrsta aktivacione funkcije koja se često koristi u logističkoj regresiji (eng. *logistic regression*). Logistička regresija je statistički model koji se koristi za predviđanje binarnih klasa, odnosno klasifikaciju uzoraka u dvije klase. Model se sastoji od linearne kombinacije ulaznih varijabli koja zatim prolazi kroz logističku funkciju, odnosno sigmoidnu funkciju. Izlaz iz sigmoidne funkcije se kreće između 0 i 1, što se može tumačiti kao vjerovatnoća da uzeti uzorak pripada određenoj klasi. Grafik sigmoidne funkcije prikazan je na slici 2.3.



Slika 2.3: Sigmoid aktivaciona funkcija

Sigmoidna funkcija je glatka i kontinualna funkcija, što znači da njen izlazni signal se mijenja postepeno, a ne naglo kao kod step funkcije, i neprekidno sa promjenom ulaza. Ove karakteristike su veoma važne za primjenu u kontekstu klasifikacije i predviđanja, jer male promjene u ulazu trebaju proizvesti male promjene u izlazu, što može dovesti do bolje klasifikacije ili predviđanja. Takođe, ova funkcija je diferencijabilna u svakoj tački, što omogućava primjenu algoritama optimizacije koji zahtijevaju izračunavanje gradijenta funkcije gubitka u procesu učenja.

Ipak, sigmoidna funkcija ima svoja ograničenja, koja su uglavnom povezana sa pojavama nestajućih gradijenata (eng. *vanishing gradients*) i eksplodirajućih gradijenata (eng. *exploding gradients*). Nestajući gradijenti nastaju kada su vrijednosti gradijenata preblizu nuli, što uzrokuje da se težine ne mijenjaju tokom treninga, čime se usporava ili čak zaustavlja proces učenja. U slučaju sigmoidne funkcije, maksimalna vrijednost prvog izvoda je 0.25, a minimalna 0, što znači da gradijenti mogu biti vrlo mali. Sigmoidna funkcija teži da „sabije“ veoma velike ili veoma male ulazne vrijednosti u blizinu svojih granica (0 ili 1), što može uzrokovati zasićenje aktivacije neurona. Kada se gradijenti približe nuli, težine neurona se ažuriraju vrlo sporo ili gotovo da se ne mijenjaju tokom procesa učenja.

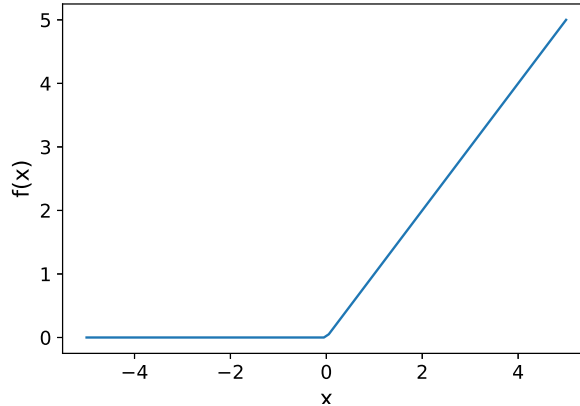
To može uzrokovati probleme kao što su sporije učenje i nemogućnost mreže da nauči složene zavisnosti. Za neuralne mreže sa samo nekoliko slojeva (eng. *shallow networks*), ovo nije veliki problem. Međutim, za DNN mreže, ukoliko se u svakom sloju izabere sigmoidna aktivaciona funkcija, to može znatno usporiti učenje i uticati na performanse modela. Nasuprot tome, pojava eksplozirajućih gradijenata se javlja kada gradijenti postanu izuzetno veliki tokom prolaska kroz mrežu. To se obično dešava kada su težine mreže postavljene na prevelike vrijednosti ili kada se multipliciraju gradijenti kroz duboke slojeve mreže. Ovo može dovesti do prevelikih koraka u optimizacionom algoritmu koji se koristi za podešavanje parametara mreže, što može uzrokovati nestabilno ponašanje mreže i loše performanse modela. Osim aktivacione funkcije, na pojave nestajućih i eksplozirajućih gradijenata može uticati i način na koji se inicijalizuju početni parametri mreže. Utvrđeno je da nasumična inicijalizacija parametara mreže može znatno ublažiti pojavu ekstremnih vrijednosti gradijenata [17].

Najjednostavnije rješenje za prevazilaženje ovih problema u DNN mrežama je da se sigmoid funkcija zamijeni nekom drugom aktivacionom funkcijom. U DL oblasti, jedna od najpopularnijih aktivacionih funkcija je **ReLU (eng. *Rectified Linear Unit*) funkcija**. Matematički zapis ReLU aktivacione funkcije je:

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (2.7)$$

Dakle, ako je vrijednost ulaza x pozitivna, ReLU aktivaciona funkcija će vratiti istu vrijednost kao ulaz. Međutim, ako je ulazna vrijednost x negativna ili nula, ReLU funkcija će dati nulu na izlazu. ReLU funkcija je, dakle, linearna funkcija za pozitivne vrijednosti, a za negativne vrijednosti vraća nulu. Bitna karakteristika ReLU aktivacione funkcije vezana je za prvi izvod ove funkcije, koji je definisan kao 1 za ulaze koji su veći od nule i kao 0 za ulaze koji su negativni. Za razliku od sigmoidne funkcije, koja ima sklonost ka zasićenju i malom gradijentu za velike vrijednosti ulaza, ReLU aktivaciona funkcija ima konstantan gradijent jednak jedinici za pozitivne vrijednosti. Ovo je ključna karakteristika koja omogućava efikasno propagiranje gradijenata unazad u dubokim neuralnim mrežama i pomaže u suzbijanju problema nestajućih gradijenata koji su česti kod sigmoidne funkcije. Zahvaljujući tome, ReLU funkcija je široko korišćena u dubokim neuralni mrežama jer omogućava bolje učenje i prenos informacija kroz slojeve. Dodatno, ReLU funkcija je jednostavnija za izračunavanje u odnosu na sigmoidnu funkciju, jer ne zahtijeva računanje eksponencijalne funkcije. Ovo čini neuralne mreže sa ReLU aktivacionom

funkcijom bržim za treniranje i obradu velikih skupova podataka. Grafički prikaz ReLU aktivacione funkcije dat je na slici 2.4.



Slika 2.4: ReLU aktivaciona funkcija

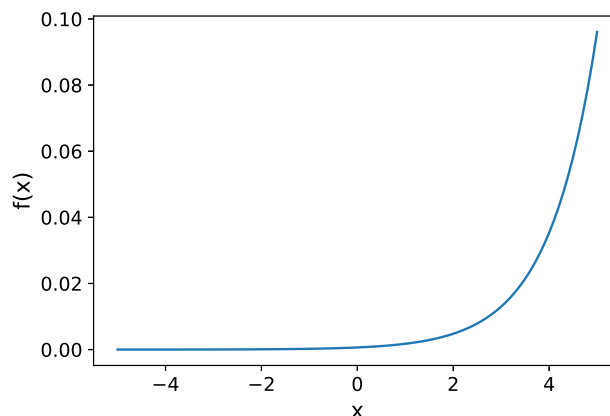
Iako je ReLU aktivaciona funkcija mnogo efikasnija od sigmoidne funkcije i rješava problem nestajućih gradijenata, ona takođe ima svoje nedostatke. Glavni problem kod ReLU aktivacione funkcije tiče se neaktivnosti neurona. Naime, kada ulazni signal ima negativnu vrijednost, ReLU funkcija će dati izlaz 0, što znači da neuron nije aktivan. Može se desiti da se veliki broj neurona deaktivira i prestane doprinositi učenju modela, što može dovesti do smanjenja kapaciteta neuralne mreže i naravno performansi, jer mreža ne može naučiti složenije obrasce u podacima. Ovaj problem se često rješava primjenom alternativnih aktivacionih funkcija koje se mogu nositi s negativnim vrijednostima ulaza, kao što su Leaky ReLU, ELU i druge.

U neuralnim mrežama koje se koriste za klasifikaciju posebno je važna **softmax aktivaciona funkcija**. U kontekstu klasifikacije, cilj neuralne mreže je da za svaki ulazni uzorak predvidi vjerovatnoću pripadanja različitim klasama. Izlazni sloj ANN mreže se sastoji od neurona, pri čemu se svaki neuron odnosi na jednu klasu koju je potrebno predvidjeti. Funkcija *softmax* se primjenjuje na vektor izlaza neurona kako bi se dobila vjerovatnoća pripadanja svakoj klasi. Matematički, *softmax* aktivaciona funkcija se može predstaviti jednačinom:

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}, \quad (2.8)$$

gdje je K broj klasa, x_i ulazni signal za i -ti neuron. Funkcija *softmax* prima x_i kao ulaz i transformiše ga u broj koji predstavlja vjerovatnoću pripadanja uzorka i -toj klasi. Da-

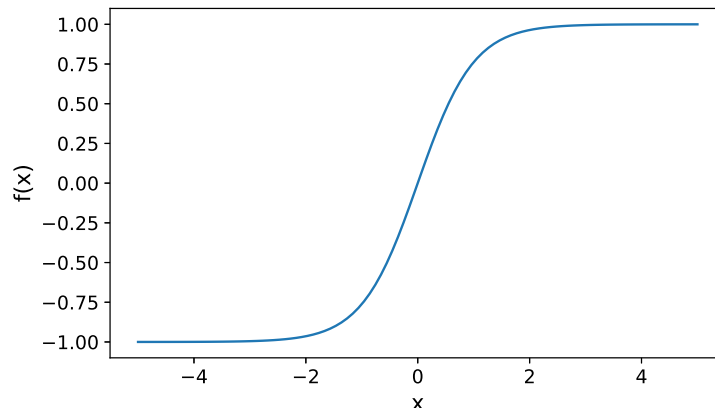
kle, ova aktivaciona funkcija uzima vektor izlaza neurona i primjenjuje eksponencijalnu funkciju na svaki element vektora kako bi se dobili nenegativni brojevi. Zatim se dobijeni vektor dijeli sa zbirom svih njegovih elemenata, što rezultira vektorom koji ima elemente u rasponu od 0 do 1 i suma svih elemenata je jednaka jedinici. Ovi elementi se tumače kao vjerovatnoća pripadanja svakoj klasi. Klasa sa najvećom vjerovatnoćom se tada smatra predviđenom klasom za dati ulazni uzorak. Grafik sa slike 2.5 odnosi se na aktivacionu funkciju *softmax*.



Slika 2.5: Softmax aktivaciona funkcija

Uočava se sličnost između grafika sa slike 2.3 koji predstavlja sigmoidnu funkciju i grafika sa slike 2.5 koji se odnosi na funkciju *softmax*. Može se smatrati da je funkcija *softmax* proširena verzija sigmoidne funkcije jer se koristi za klasifikaciju sa više klasa, za razliku od sigmoidne funkcije koja se koristi za binarnu klasifikaciju. Takođe, sigmoidna aktivaciona funkcija se obično koristi u skrivenim slojevima mreže za transformaciju linearnog izlaza u nelinearni, dok se funkcija *softmax* koristi u izlaznom sloju za transformaciju linearnog izlaza u vjerovatnoće za svaku klasu u problemu klasifikacije. *Softmax* funkcija je stoga često povezana sa kategoričkim izlazima, dok se sigmoidna funkcija može koristiti za bilo koju vrstu kontinuiranog izlaza. Pored funkcije *softmax*, postoje i još neke varijante sigmoidne funkcije poput **tanh aktivacione funkcije** (hiperbolički tangens) koja takođe ima karakterističan S oblik kao sigmoid funkcija, ali ima izlaz koji se kreće između -1 i 1 .

Grafik sa slike 2.6 predstavlja tanh aktivacionu funkciju. Uočava se da što je veći ulaz (pozitivniji), to će izlazna vrednost biti bliža 1, dok što je manji ulaz (negativniji), to će izlaz biti bliži -1 .



Slika 2.6: Tanh aktivaciona funkcija

Obično se aktivaciona funkcija *tanh* koristi u skrivenim slojevima NN mreže jer se njene vrijednosti nalaze između između -1 i 1 , stoga, srednja vrijednost aktivacije za skriveni sloj ispada nula ili veoma blizu nuli, što olakšava treniranje mreže i ima efekt centriranja podataka. Kada se centriraju podaci, to znači da se prosječna vrijednost podataka pomijera prema nuli što čini učenje za sljedeći sloj lakšim [18]. Uglavnom, funkcija *tanh* je bolja opcija za skrivene slojeve u neuralnim mrežama, dok se sigmoidna funkcija češće koristi za izlazni sloj, posebno za binarnu klasifikaciju.

Može se zaključiti da postoji veliki broj aktivacionih funkcija jer različiti problemi i arhitekture neuralnih mreža zahtijevaju različite karakteristike funkcija. Uopšteno govoreći, uloga aktivacione funkcije u ANN mreži je da dodaje nelinearnost u izračunavanje izlaza neurona. Dakle, aktivaciona funkcija omogućava neuralnoj mreži da nauči kompleksne funkcije koje su nelinearne, što je ključno za rješavanje mnogih problema u različitim oblastima, kao što su prepoznavanje slika, prevođenje prirodnog jezika, klasifikacija podataka i još mnogo toga. Osim toga, aktivaciona funkcija pomaže u izbjegavanju problema u kojima se težinski koeficijenti zaglave u lokalnom minimumu funkcije gubitka, što bi otežalo učenje neuralne mreže. Aktivaciona funkcija takođe može pomoći u smanjenju pretjeranog uticaja nekih težinskih koeficijenata na izlaz neurona, što bi inače moglo da dovede do pretreniranosti (eng. *overfitting*) modela, slabe generalizacije i loših performansi na novim podacima. Evidentno je da odabir aktivacione funkcije predstavlja važan faktor za postizanje dobrih performansi u višeslojnim neuralnim mrežama.

2.1.2 Funkcija gubitka

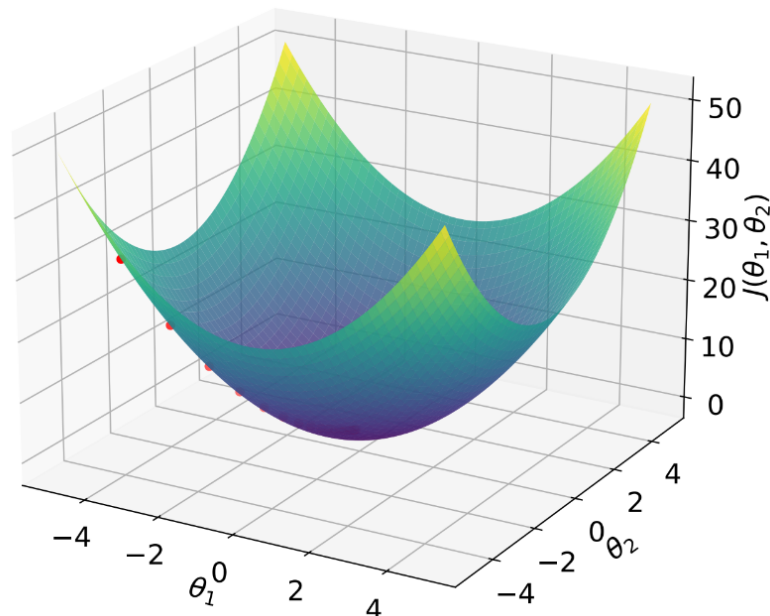
Vratimo se sada na logističku regresiju. U matematičkoj notaciji, izlaz iz modela logističke regresije može se izraziti kao:

$$\hat{y} = \sigma(W^T \cdot X + b), \quad (2.9)$$

gdje je σ sigmoidna aktivaciona funkciju. Cilj je odrediti parametre W i b tako da se minimizuje odstupanje između \hat{y} , vrijednosti izlaza koju je model predvidio, i ciljne varijable y koja predstavlja stvarnu vrijednosti izlaza. Matematička funkcija koja izražava odstupanje između stvarnih i predviđenih izlaza neuralne mreže za određeni skup podataka je funkcija gubitka. Za slučaj logističke regresije, funkcija gubitka se najčešće računa kao srednja kvadratna greška između stvarne i procijenjene vrijednosti. Dakle, u k -toj iteraciji treninga funkcija gubitka se računa na sljedeći način:

$$J_k = \frac{1}{2} (\hat{y} - y)^2 = \frac{1}{2} (\sigma(W_k^T X) - y)^2. \quad (2.10)$$

Ispostavlja se da je funkcija gubitka u slučaju logističke regresije konveksna funkcija, što pokazuje grafik sa slike 2.7.



Slika 2.7: Funkcija gubitka u logističkoj regresiji

Određivanje minimuma funkcije gubitka, odnosno vrijednosti parametara modela koji će dati najmanju grešku predikcije vrši se pomoću iterativnog optimizacionog algoritama koji se bazira na gradijentu funkcije gubitka. Metod gradijentnog spuštanja (eng. *gradient descent*) počinje sa inicijalizacijom parametara modela na neku početnu vrijednost. Za logističku regresiju, skoro svaki metod inicijalizacije funkcionira iz razloga što funkcija gubitka nema lokalnih minimuma. Dakle, bez obzira na početnu tačku, odnosno početne vrijednosti parametara, iterativno pomjeranje u smjeru negativnog gradijenta funkcije gubitka dovešće do globalnog minimuma ove funkcije. Ažuriranje parametara modela se vrši na sljedeći način:

$$W_{k+1} = W_k - \mu \frac{\partial J_k}{\partial W_k} = W_k - \mu (\hat{y}_k - y) \frac{\partial \hat{y}_k}{\partial W_k} X = W_k - \mu (\hat{y}_k - y) \hat{y}'_k X. \quad (2.11)$$

Na osnovu gornje jednačine vidimo da je izvod aktivacione funkcije \hat{y}'_k bitan za ažuriranje parametara modela u neuralnoj mreži, jer se on koristi kako bi se izračunao gradijent funkcije gubitka u odnosu na težine i *bias*. Takođe, na osnovu znaka minus u jednačini 2.11, zaključujemo da se parametri ažuriraju u suprotnom smjeru gradijenta funkcije gubitka. Gradijent funkcije gubitka je vektor čiji su elementi parcijalni izvodi funkcije gubitka u odnosu na svaki parametar u modelu. Smjer suprotan gradijentu pokazuje smjer najbržeg smanjenja funkcije gubitka, pa se parametri ažuriraju u tom smjeru kako bi se postigao minimum funkcije gubitka. Dakla, iz iteracije u iteraciju, prave se koraci u pravcu negativnog gradijenta funkcije gubitka a veličinu koraka određuje parametar μ koji se naziva stopa učenja (eng. *learning rate*). Generalno, što je stopa učenja veća, to će se model brže prilagođavati i postizati bolje performanse, ali će biti veća šansa da se preskoči globalni minimuma funkcije gubitka, što može dovesti do nepostizanja optimalnog rješenja. Sa druge strane, preniska stopa učenja može značiti da će algoritam konvergirati veoma sporo ili uopšte neće konvergirati. Dakle, stopu učenja treba pažljivo odabrati tako da se izbjegnu ove dvije ekstremne situacije. Optimalna stopa učenja zavisi od mnogih faktora, kao što su veličina trening skupa podataka, veličina parametara modela i specifičnosti funkcije gubitka koja se minimizira. Jedan od načina da se pronađe optimalna stopa učenja je da se eksperimentalno testiraju različite vrijednosti i provjeri koji korak daje najbolje performanse na validacionom skupu podataka.

Postoje različiti tipovi funkcija gubitka, a izbor odgovarajuće funkcije zavisi od vrste problema i vrste izlaza koji se predviđa. Na primjer, funkcija srednjih kvadrata (eng. *Mean Square Error* - MSE) se obično koristi za regresione probleme, dok se kategorič-

ka unakrsna entropija (eng. *categorical cross-entropy*) koristi za klasifikacione probleme. S obzirom na to da je procjena brzine vozila posmatrana kao regresioni i klasifikacioni problem, matematički izrazi za pomenute funkcije gubitka su dati u nastavku.

MSE se računa kao prosječna kvadratna razlika između stvarnih (y_i) i predviđenih (\hat{y}_i) vrijednosti izlazne promenljive dužine n . Jednačina 2.12 izrazava matematičku formulu za MSE:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (2.12)$$

Kategorička unakrsna entropija je funkcija gubitka koja se često koristi u klasifikacionim problemima sa više klasa, kada se izlazi tumače kao vjerovatnoća pripadnosti određenoj klasi. Matematička formula za kategoričku unakrsnu entropiju data je jednačinom 2.13:

$$\text{CE} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{ij} \log(p_{ij}), \quad (2.13)$$

gdje N označava broj instanci, C označava broj klasa, y_{ij} predstavlja binarnu oznaku koja je jednaka jedinici ako je instanca i pripadnik klase j , u suprotnom je nula, dok p_{ij} predstavlja vjerovatnoću koju model dodjeljuje instanci i za klasu j .

Važno je napomenuti da je izbor funkcije gubitka direktno povezan sa aktivacionom funkcijom koja se koristi u neuralnoj mreži. Na primjer, za regresione probleme, kada je izlaz neuralne mreže kontinualna vrijednost, često se koristi linearna aktivaciona funkcija u izlaznom sloju i MSE funkcija gubitka. S druge strane, za klasifikacione zadatke, ako se koristi *softmax* aktivaciona funkcija za višeklasnu klasifikaciju, tada se kao funkcija gubitka može koristiti kategorička unakrsna entropija.

2.1.3 Procjena uspješnosti modela

U mašinskom učenju, uobičajeno je da se skup podataka podijeli na tri dijela: trening skup, validacioni skup i test skup. Trening skup se koristi za treniranje modela, odnosno za podešavanje parametara modela na osnovu podataka koji su mu dostupni. Međutim, kada se model trenira samo na trening skupu, on može postići visoku preciznost i performanse na tim podacima, ali može se desiti da nema dobre performanse na novim podacima. To

je zato što se model prilagođava trening podacima u potpunosti i može zapamtiti karakteristike trening podataka, umjesto da nauči da generalizuje ta znanja i primijeni ih na nove podatke. Stoga se obično koristi validacioni skup podataka, kako bi se procijenio kvalitet modela u fazi treniranja i detektovala pretreniranost. Ako model ima visoku preciznost na trening skupu, ali nisku preciznost na validacionom skupu, to može ukazati na prenaučenosť i potrebu za prilagođavanjem parametara modela kako bi se poboljšale performanse na novim podacima. Treniranje neuralne mreže ima za cilj da nauči mrežu da generalizuje i donosi tačne predikcije na novim podacima, umjesto da samo memoriše ulaze i izlaze iz skupa podataka koji su korišćeni za treniranje. Uvođenjem validacionog skupa podataka obezbjeđuje da model bude sposoban da rješava probleme koji se javljaju u stvarnim situacijama, a ne samo one koji su prisutni u skupu podataka za treniranje.

Konačno, nakon što se model trenira i testira na trening i validacionom skupu, evaluacija konačnih performansi modela se vrši na test skupu, koji se sastoji od podataka koji nisu korišćeni za obuku. Test skup se koristi za evaluaciju performansi odabranog modela i daje konačnu procjenu performansi modela na novim podacima. Performanse klasifikacionog modela mogu se vizualizovati kroz matricu konfuzije (eng. *confusion matrix*) koja daje detaljan uvid u rezultate klasifikacije za svaku klasu jer sadrži broj tačno klasifikovanih i pogrešno klasifikovanih instanci za svaku klasu problema. U binarnoj klasifikaciji moguće su četiri različite kombinacije rezultata klasifikacije u odnosu na stvarne klase i to su:

- Stvarno pozitivni - SP: Broj instanci koje su tačno klasifikovane kao pozitivne.
- Lažno pozitivni - LP: Broj instanci koje su greškom klasifikovane kao pozitivne, iako pripadaju negativnoj klasi.
- Lažno negativni - LN: Broj instanci koje su greškom klasifikovane kao negativne, iako pripadaju pozitivnoj klasi.
- Stvarno negativni - SN: Broj instanci koje su tačno klasifikovane kao negativne.

Matrica konfuzije za binarnu klasifikaciju može se predstaviti na sljedeći način:

	Stvarna pozitivna klasa	Stvarna negativna klasa
Predviđena pozitivna klasa	SP	LP
Predviđena negativna klasa	LN	SN

Dakle, svaka klasa koja se koristi u procesu klasifikacije ima svoj red i svoju kolonu u matrici konfuzije. Polja u koloni predstavljaju broj instanci koje je klasifikacioni model dodijelio određenoj klasi, dok polja u redu predstavljaju broj instanci koje zaista pripadaju toj klasi. Elementi na dijagonali matrice predstavljaju broj ispravno klasifikovanih instanci.

U odnosu na matricu konfuzije moguće je definisati različite metrike za procjenu kvaliteta klasifikacije i to su:

- Tačnost (eng. *accuracy*): Tačnost predstavlja udio tačno klasifikovanih instanci u odnosu na ukupan broj instanci. Ova metrika je pogodna kada su sve klase podjednako važne. Formula za tačnost je:

$$\text{Tačnost} = \frac{SP + SN}{SP + SN + LP + LN} \quad (2.14)$$

- Preciznost (eng. *precision*): Preciznost mjeri koliko su tačno pozitivne instance identifikovane među onima koje su klasifikovane kao pozitivne. Preciznost se koristi kada je fokus na minimiziranju lažno pozitivnih rezultata. Formula za preciznost je:

$$\text{Preciznost} = \frac{SP}{SP + LP} \quad (2.15)$$

- Odziv (eng. *recall*): Odziv mjeri koliko su tačno pozitivne instance identifikovane u odnosu na ukupan broj stvarno pozitivnih instanci. Odziv se koristi kada je fokus na minimiziranju lažno negativnih rezultata. Formula za odziv je:

$$\text{Odziv} = \frac{SP}{SP + LN} \quad (2.16)$$

- F1 mjera (eng. *F1 score*): F1-mjera je harmonijska sredina između preciznosti i odziva. Ova metrika uzima u obzir i preciznost i odziv kako bi pružila balansiranu mjeru performansi klasifikatora. Formula za F1-mjeru je:

$$\text{F1-mjera} = 2 \cdot \frac{\text{Preciznost} \cdot \text{Odziv}}{\text{Preciznost} + \text{Odziv}} \quad (2.17)$$

U problemima regresije, neke od često korišćenih metrika su:

- Srednja apsolutna greška (eng. *Mean Absolute Error* - MAE): Ovo je srednja vrijednost apsolutnih vrijednosti razlika između stvarnih i predviđenih vrijednosti.
- Korijen srednje kvadratne greška (eng. *Root Mean Squared Error* - RMSE): Ovo je korijen srednje vrijednosti kvadrata razlika između stvarnih i predviđenih vrijednosti.

2.2 Duboko učenje

Duboko učenje je u posljednjih nekoliko godina postalo jedno od najpopularnijih oblasti u mašinskom učenju i vještačkoj inteligenciji. Tehnike dubokog učenja su u pozadini proizvoda koje svakodnevno koristimo kao što su digitalni asistenti, sistemi za prepoznavanje slika, prevođenje jezika i davanje preporuka. Duboko učenje se oslanja na koncept dubokih neuralnih mreža, koje su složeni matematički modeli kreirani sa namjerom da simuliraju način na koji ljudski mozak obrađuje informacije. Duboke neuralne mreže sastoje se od velikog broja slojeva, gdje svaki sloj mreže obavlja nelinearnu transformaciju nad ulaznim podacima, a izlaz tog sloja se koristi kao ulaz u sledeći sloj. Kako se podaci kreću kroz sve više slojeva, složenost nelinearnih transformacija se povećava, što dovodi do konstrukcije sve složenijih i apstraktnijih reprezentacija podataka. Može se reći da se ključna prednost dubokog učenja ogleda u sposobnosti ovih modela da nauče bitne karakteristike iz podataka. U konvencionalnim metodama mašinskog učenja potrebno je ručno izabrati karakteristike koje najbolje opisuju zadati problem i zatim ih koristiti u modelu za klasifikaciju ili regresiju. Međutim, u dubokom učenju, vještačka neuralna mreža je u stanju da sama nauči relevantne karakteristike iz podataka, što eliminiše potrebu za angažovanjem domenskog eksperta, štedi na vremenu i smanjuje greške koje nastaju pod uticajem ljudskog faktora.

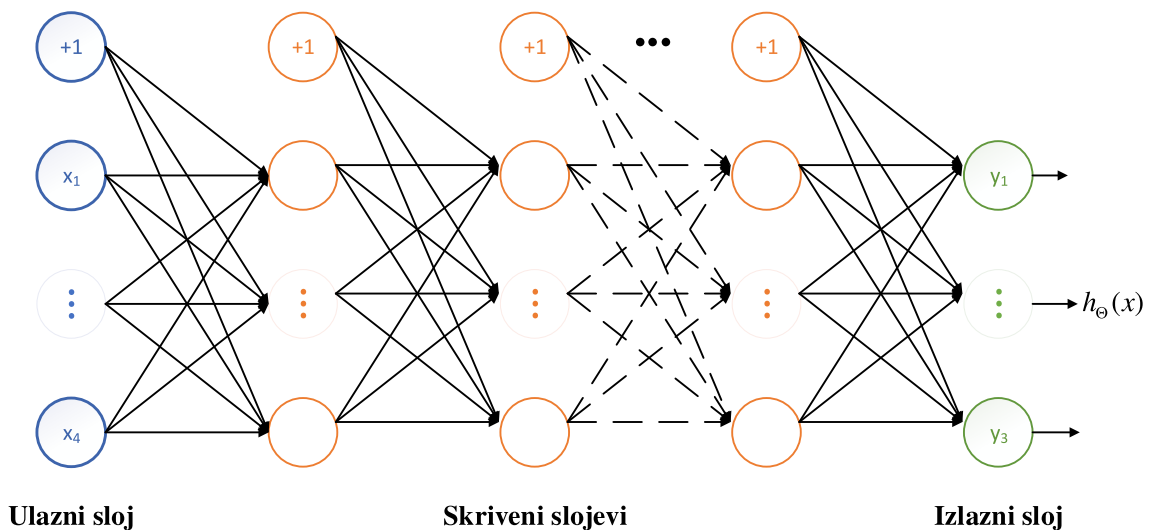
Duboko učenje se obično povezuje sa velikim skupovima podataka, gdje su podaci neobrađeni, odnosno sirovi (eng. *raw*), uglavnom neoznačeni i nekategorisani. Zahvaljujući distribuiranom predstavljanju podataka modeli dubokog učenja mogu da nauče kompaktne i apstraktne karakteristike koje se potom mogu koristiti za predviđanje novih uzoraka, čime se postiže bolja generalizacija modela. Naime, umjesto da se ulazni podaci predstavljaju u originalnom obliku, distribuirano prikazivanje podataka podrazumijeva pretvaranje ulaznih podataka u niz apstraktnih karakteristika, što omogućava da se svaki uzorak prikaže na kompaktniji način. Kada se jednom nauče karakteristike koje opisuju neke uzorke, moguće ih je iskoristiti za opisivanje i novih uzoraka, što se postiže kombinovanjem karakteristika na nove načine i stvaranjem novih konfiguracija koje opisuju nove uzorke, koje model ranije nije vidio [19]. Osim toga, važna prednost apstraktnijih reprezentacija je da mogu biti invarijantne na lokalne promjene u ulaznim podacima. Invarijantnost karakteristika je posebno poželjna u zadacima kompjuterske vizije, gdje se objekti mogu pojaviti u različitim oblicima, veličinama, pozicijama i orijentacijama. Invarijantne karakteristike omogućavaju mreži da prepozna objekte bez obzira na te varijacije

u ulaznim podacima. Pokazano je da sa povećanjem dubine neuralne mreže, reprezentacije koje su naučene mogu imati i povećan stepen invarijantnosti. Naime, invarijantnosti na manje promjene u podacima uče se u nižim slojevima mreže, a zatim se kombinuju u višim slojevima kako bi se postepeno formirale sve invarijantnije karakteristike [20].

Učenje u dubokim neuralnim mrežama odvija se kroz postupak optimizacije parametara mreže kako bi se minimizovala funkcija gubitka koja se koristi za procjenu greške modela. Prvo, se izvršava takozvana **propagacija unaprijed** (eng. *forward propagation*) koja omogućava da se informacija obrađuje kroz slojeve i da se na kraju dobije konačni rezultat mreže. Zatim, izlazni rezultat mreže se poredi sa stvarnom vrijednosti ciljne varijable, a funkcija gubitka se koristi za procjenu greške. Nakon toga, slijedi proces **propagacije unazad** (eng. *backward propagation*) koji se koristi za izračunavanje gradijenta funkcije gubitka u odnosu na parametre mreže kako bi se minimizovala greška modela. Procesi propagacije unaprijed i unazad su osnovni koncepti u neuralnim mrežama stoga će isti biti detaljno opisani na narednim stranicama.

2.2.1 Propagacija unaprijed i unazad

U dubokim neuralnim mrežama, propagacija unaprijed je proces kojim se ulazni podaci propuštaju, počevši od ulaznog sloja, preko skrivenih slojeva, do izlaznog sloja, kako bi se generisale izlazne vrijednosti. Uzmimo u obzir duboku neuralnu mrežu sa L slojeva, datu na na slici 2.8.



Slika 2.8: Duboka neuralna mreža

Kada se ulazni podaci prvi put propuste kroz mrežu, vrijednosti neurona u prvom sloju se računaju množenjem težinskih koeficijenata sa ulaznim podacima, a zatim se primjenom aktivacione funkcije na ove vrijednosti dobijaju izlazi neurona iz prvog sloja. Generisane izlazne vrijednosti iz prvog sloja zatim se koriste kao ulazni podaci za drugi sloj, a ovaj proces se ponavlja sloj po sloj, za sve skrivene slojeve u mreži, do izlaznog sloja. Izlazna vrijednost iz DNN mreže se dobija primjenom aktivacione funkcije na izlazne vrijednosti poslednjeg sloja.

Uvedimo sada oznake koje ćemo koristiti za definisanje propagacije unaprijed. Neka je $a_k^{[l]}$ vrijednost aktivacione funkcije u l -tom sloju i k -tom neuronu, koja se još naziva i aktivacija neurona. Zatim, neka se $z_k^{[l]}$ odnosi na težinsku sumu ulaznih vrijednosti u k -ti neuron na koje se dodaje sistemska greška modela u l -tom sloju, na sljedeći način:

$$z_k^{[l]} = W_k^{[l]T} a_k^{[l-1]} + b_k^{[l]}. \quad (2.18)$$

Izlazna vrijednost iz DNN mreže se dobija primjenom aktivacione funkcije g na izlazne vrijednosti posljednjeg sloja. Odnosno, možemo pisati da je:

$$\hat{y} = a^{[L]} = g^{[L]}(z_k^{[L]}). \quad (2.19)$$

Dakle, izlaz iz duboke neuralne mreže se računa primjenom aktivacione funkcije na težinsku sumu neurona u poslednjem sloju mreže, koja je dobijena propuštanjem ulaznih podataka kroz sve prethodne slojeve. Proces propagacije unaprijed je bitan korak u trenin-gu neuralne mreže, jer omogućava da se dobiju predikcije na osnovu ulaznih podataka, što je potrebno kako bi se izračunala funkcija gubitka i vršilo podešavanje parametara modela tokom procesa učenja.

Nakon propagacije unaprijed, slijedi proces propagacije unazad koji se koristi za računanje gradijenata funkcije gubitka u odnosu na parametre neuralne mreže. Dobijeni gradijenti se zatim koriste za ažuriranje parametara mreže korištenjem nekog optimizaci-onog algoritma, kao što je na primjer metod gradijentnog spuštanja koji je pomenut kod logističke regresije. Propagacija unazad počinje tako što se prvo se izračunaju gradijent funkcije gubitka u odnosu na izlazni vektor $a^{[L]}$, odnosno:

$$da^{[L]} = \frac{\partial J}{\partial a^{[L]}} \quad (2.20)$$

Zatim se gradijent $da^{[L]}$ koristi za izračunavanje gradijenta funkcije gubitka u odnosu na težinsku sumu $z^{[L]}$ u posljednjem sloju:

$$dz^{[L]} = \frac{\partial J}{\partial z^{[L]}} = \frac{\partial J}{\partial a^{[L]}} \cdot \frac{\partial a^{[L]}}{\partial z^{[L]}} = da^{[L]} \cdot g^{[L]'}(z^{[L]}). \quad (2.21)$$

Potom, koristeći taj gradijent, možemo izračunati gradijent funkcije gubitka u odnosu na izlazne vrijednosti iz prethodnog sloja $a^{[L-1]}$, što označavamo sa $da^{[L-1]}$, na sledeći način:

$$da^{[L-1]} = W^{[L]T} dz^{[L]}. \quad (2.22)$$

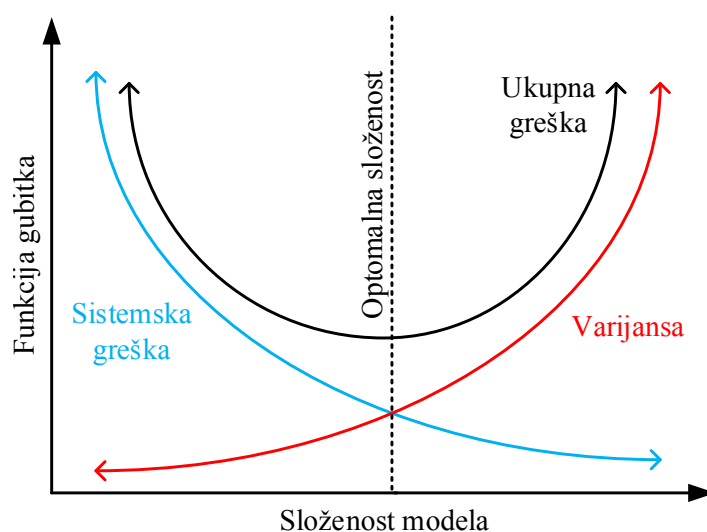
Konačno, gradijent funkcije gubitka u odnosu na parametre $W^{[L]}$ se izračunava kao:

$$dW^{[L]} = \frac{\partial J}{\partial W^{[L]}} = dz^{[L]} \cdot a^{[L-1]T}. \quad (2.23)$$

Treba naglasiti da su relacije (2.20) - (2.23) prikazane u vektorskoj formi, te da „“ označava matrično množenje, a „ T “ označava transponovanje. Na ovaj način, umjesto nad jednim neuronom, prikazano je izvršavanje procesa nad svim neuronima u određenom sloju mreže. Vektorske jednačine na taj način dovode do značajnog ubrzanja izvršavanja postupka propagacije unaprijed i unazad, što je posebno važno za duboke neuronske mreže sa velikim brojem neurona i slojeva. Proces propagacije unaprijed i unazad se obično ponavlja kroz više epoha kako bi se parametri neuralne mreže postepeno prilagođavali ulaznim podacima. Na kraju procesa učenja, neuralna mreža bi trebala biti u stanju da sa visokom tačnošću klasifikuje ili predviđa izlaze na osnovu ulaznih podataka.

U dubokom učenju gdje se modeli obično treniraju kroz velik broj epoha, funkcija gubitka se može koristiti za praćenje performansi modela tokom trening procesa. S tim u vezi bitno je uvesti dva pojma koji se nazivaju sistemska greška i varijansa (eng. *variance*) modela. U dubokom učenju, pojam sistematske greške i varijanse odnosi se na balans između pristrasnosti i varijabilnosti modela. Kada je model pristrasan, to znači da je njegova fleksibilnost ograničena i nije u stanju da nauči relevantne obrasce u podacima. U suštini, ovo znači da je model previše jednostavan i nedovoljno fleksibilan da bi se prilagodio podacima. Visoka pristrasnost može dovesti do podtreniranosti (eng. *underfitting*). U ovom slučaju, model ne postiže dobre rezultate ni na trening skupu, ni na novim podacima. Varijansa modela, sa druge strane, predstavlja problem koji se javlja kada je model prekomplikovan i previše fleksibilan u odnosu na podatke za treniranje. U ovom slučaju, model može da nauči ne samo relevantne obrasce, već i šum u podacima. Visoka

varijansa može dovesti do pretreniranosti, što znači da model može jako dobro da se prilagodi podacima za treniranje, ali loše se generalizuje na nove podatke. Da bi se postigao dobar balans između pristrasnosti i varijabilnosti modela, potrebno je pravilno odabrati složenost modela i primijeniti tehnike regularizacije. Takođe, važno je pratiti funkciju gubitka na validacionom skupu tokom treninga kako bi se pravovremeno prepoznali problemi sa sistemskom greškom ili varijansom i unijele odgovarajuće korekcije. Na slici 2.9 prikazana je zavisnost složenosti modela i funkcije gubitka.



Slika 2.9: Funkcija gubitka za procjene performansi modela

Dakle, ukoliko vrijednost funkcije gubitka opada tokom treninga, to bi moglo ukazivati na to da se neuralna mreža uspješno prilagođava trenutnom skupu podataka. Međutim, ako se funkcija gubitka ne smanjuje tokom treniranja, to može ukazivati na problem koji se odnosi na veliku sistemsku grešku. Dakle, u tom slučaju model nije dovoljno sposoban da nauči prave zakonitosti u podacima. U ovom slučaju, treba razmotriti dodavanje novih slojeva ili povećanje broja neurona po sloju kako bi se povećala složenost modela i poboljšale performanse. Ukoliko funkcija gubitka opada a zatim naglo počne da raste poslije izvjesnog broja epoha, to može ukazivati na problem pretreniranosti modela, odnosno da se model previše prilagođava skupu za treniranje. Upravo ovo dovodi do povećanja varijanse modela, tj. do smanjenja sposobnosti modela da dobro predviđa na novim podacima. Optimalna složenost modela je prikazana isprekidanom linijom.

2.2.2 Regularizacija

Duboke neutralne mreže se često suočavaju sa izazovom pretreniranosti, koji je već ranije opisan kao sklonost mreže da se prilagođava trening skupu podataka do te mjere da gubi sposobnost generalizacije na nove podatke. Pretreniranost se dešava kada neuralna mreža postane pretjerano specifična za podatke na kojima uči i gubi sposobnost generalizacije na nove podatke. Jedan od razloga zašto se pretreniranje dešava je nedovoljna količina dostupnih podataka. Ako mreža ima previše parametara u odnosu na broj dostupnih podataka, ona može da nauči sve primjere iz trening skupa. Takođe, ako se skup za obučavanje sastoji od sličnih primjera, mreža će naučiti samo jedan skup obrazaca, a neće biti u stanju da generalizuje na druge primjere. Pretreniranje se može izbjeći primjenom regularizacionih tehnika. Postoje različite metode regularizacije koje se koriste za smanjenje prenaučivosti mreže, u ovoj podsekciji biće spomenute najpopularnije.

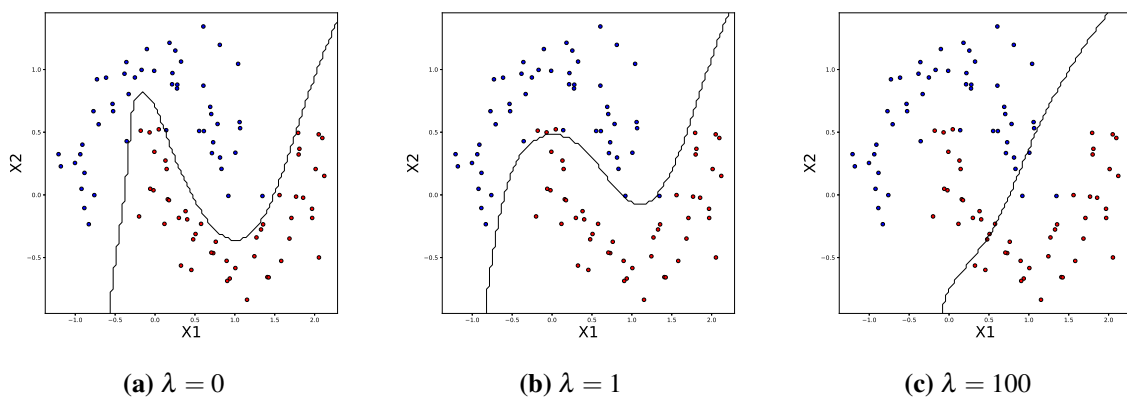
Dvije jednostavne i vrlo popularne metode regularizacije u dubokom učenju su **L1** i **L2 regularizacija**. L1 regularizacija (eng. *lasso regression*) podrazumijeva da se funkciji gubitka dodaje kazneni član koji se sastoji od parametra regularizacije λ pomnoženog s sumom apsolutnih vrijednosti težinskih koeficijenata neuralne mreže. Na ovaj način neuralna mreža, osim funkcije gubitka, minimizuje i ukupnu sumu apsolutnih vrijednosti težinskih koeficijenata. Dakle, L1 regularizacija postavlja ograničenje na vrijednosti težinskih koeficijenata i na taj način naglašava bitne karakteristike dok istovremeno uklanja manje važne karakteristike postavljanjem vrijednosti njihovih koeficijenata na nulu. Ovo smanjuje složenost modela i smanjuje vjerovatnoću prenaučivosti, što poboljšava sposobnost modela da se generalizuje na novim primjerima. Funkcija greške na koju je primijenjena L1 regularizacija data je sljedećom jednačinom:

$$J_k = \frac{1}{2}(\hat{y} - y)^2 + \lambda \sum_{l=1}^L \left| \left| W^{[l]} \right| \right|. \quad (2.24)$$

L2 regularizacija (eng. *ridge regression*) dodaje sumu kvadrata vrijednosti težinskih koeficijenata mreže u funkciju gubitka. Glavna razlika L2 regularizacije u odnosu na L1 regularizaciju je u tome što ona vodi do smanjenja vrijednosti većeg broja koeficijenata, ali se oni i dalje razlikuju od nule. S druge strane, L1 regularizacija vodi ka postavljanju nekih manje bitnih koeficijenata na nulu, što dovodi do smanjenja dimenzionalnosti i kompleksnosti modela. Funkcija greške kod L2 regularizacije se ona može zapisati na sljedeći način:

$$J_k = \frac{1}{2}(\hat{y} - y)^2 + \lambda \sum_{l=1}^L \|W^{[l]}\|^2, \quad (2.25)$$

gdje $\|W^{[l]}\|^2$ predstavlja L2 normu matrice težinskih koeficijenata u sloju l . Parametar λ određuje nivo regularizacije i njegova vrijednost se određuje eksperimentalno na osnovu rezultata na validacionom skupu podataka. Kada je $\lambda = 0$ funkcija greške je identična slučaju bez regularizacije, dok će za velike vrijednosti parametra λ težinski koeficijenti mreže težiti nuli. Uticaj parametra regularizacije u slučaju klasifikacije prikazan je na slici 2.10.



Slika 2.10: Uticaj parametra regularizacije na granicu odlučivanja

Kada nema regularizacije, odnosno za $\lambda = 0$, klasifikator će obično ispravno klasifikovati gotovo sve primjere u skupu za obučavanje. Međutim, granica odlučivanja će biti vrlo izvijana, kao što je prikazano na slici 2.10a, što ukazuje na prenaučenosť modela. S druge strane, kada se uzme parametar regularizacije $\lambda = 1$ granica odlučivanja je jednostavnija, ali i dalje dobro razdvaja pozitivne i negativne primjere. Ovaj slučaj optimalne regularizacije je prikazan na slici 2.10b. Međutim, ako se koeficijent regularizacije postavi na previše visoku vrijednost npr. $\lambda = 100$, model se neće dobro prilagoditi podacima i granica odlučivanja neće pratiti podatke, što može dovesti do loših performansi modela, slika 2.10c.

U dubokom učenju, vrlo popularna tehnika regularizacije, je takozvano **odbacivanje neurona** (eng. *dropout*). Ovaj metod regularizacije se zasniva na nasumičnom isključivanju određenih neurona iz mreže prilikom treniranja. Ovaj postupak nasumičnog isključivanja neurona omogućava da se u svakom prolazu kroz mrežu koristi samo podskup neurona, što sprečava pretreniranje mreže tako što je prisiljava da nauči relevantne karakteristike umjesto da se prilagodi šumu ili nebitnim varijablama. Tehnika *dropout* omogu-

ćava neuronima da se prilagode nezavisno, bez obzira na ostale neurone u mreži, što utiče na bolju distribuciju naučenih karakteristika širom mreže.

Ukoliko vektor težinskih koeficijenata koji potiču od k -tog neurona u sloju l označimo sa $w_k^{[l]}$, vektor težinskih koeficijenata u sloju u kojem se koristi *dropout* može se zapisati kao:

$$\hat{w}_k^{[l]} = \begin{cases} w_k^{[l]} & , \text{ sa vjerovatnoćom } P(c) \\ 0 & , \text{ u suprotnom.} \end{cases} \quad (2.26)$$

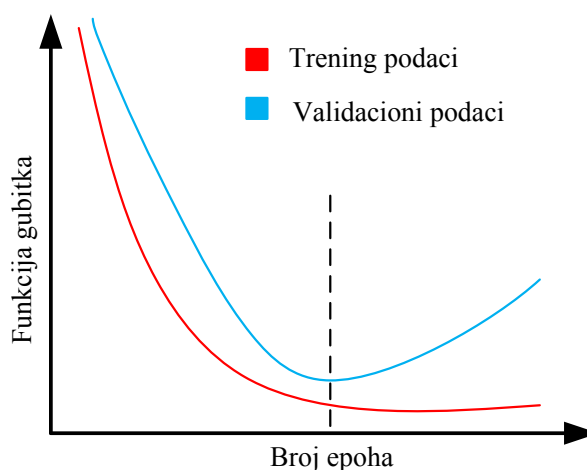
U metodi *dropout*, uvodi se novi hiperparametar c koji određuje vjerovatnoću pri kojoj se izlazni podaci sloja zadržavaju. Ukoliko je vrijednost hiperparametra c jednaka 1 onda nema odbacivanje neurona. Što je vrijednost c manja, uticaj regularizacije na mrežu je veći, odnosno odbacuje se više neurona iz određenog sloja. Ova vrijednost se ne mora podudarati za sve slojeve, već se može individualno odrediti za svaki sloj. Obično se koristi vrijednost između 0.2 i 0.5, ali se može eksperimentalno podešavati za svaki pojedinačni slučaj. Bitno je pomenuti da se *dropout* primjenjuje samo u trening fazi, dok se u fazi testiranja izlazne težine neurona koji je bio odbačen množe sa c kako bi se kompenzovao gubitak težine uzrokovan primjenom ove tehnike [21].

Tehnika *dropout* se često koristi u oblasti kompjuterske vizije, gdje je broj dostupnih podataka obično ograničen i nedovoljno velik, što može dovesti do prenaučeniosti modela. U kompjuterskoj viziji se takođe često koristi i **augmentacija podataka** (eng. *data augmentation*). Ovaj pristup se sastoji od generisanja novih primjera trening skupa iz postojećih primjera pomoću različitih transformacija, kao što su rotacija, skaliranje i promjena osvjetljenja. Augmentacija podataka povećava raznolikost trening skupa i omogućava modelu da nauči invarijantnosti na različite transformacije.

U konvolucionim neuralnim mrežama *dropout* se primjenjuje na ulaze u konvolucione slojeve i na izlaze potpuno povezanih slojeva koji slijede nakon konvolucionih slojeva. Druge tehnike regularizacije, kao što je takozvana **batch normalizacija** se češće koriste kod konvolucionih neuralnih mreža. Kada se trenira neuralna mreža, distribucija ulaza svakog sloja se mijenja kako se mijenjaju parametri prethodnih slojeva, što usporava proces učenja i čini ga teškim. Ovaj problem se u literaturi objašnjava kao unutrašnje kovarijantno pomijeranje (eng. *internal covariate shift*). Ovaj pojava se može ublažiti normalizacijom ulaza svakog sloja za svaku manju grupu podataka tokom treninga, što osigurava da će srednja vrijednost i standardna devijacija ulaza slojeva uvijek ostati iste. Na

taj način, smanjuje se količina promjene u distribuciji ulaza slojeva i dublji slojevi imaju stabilniju predstavu o tome kakve će vrijednosti ulaza biti. Pomenuti pristup omogućava korištenje većih stopa učenja, čime se ubrzava trening i u nekim slučajevima eliminiše potrebu za primjenom tehnike *dropout* [22].

Metoda ranog zaustavljanja (eng. *early stopping*) predstavlja jednostavan način da se izbjegne pretreniranost modela. Ova metoda se zasniva na praćenju performansi modela na validacionom skupu tokom procesa treniranja. Konkretno, kada se greška na skupu za validaciju počne povećavati nakon određenog broja epoha, može se pretpostaviti da je došlo do pretreniranja modela. Tada se trening prekida i uzima se model koji je imao najbolju performansu na validacionom skupu tokom procesa treniranja. Grafik sa slike 2.11 opisuje metod ranog zaustavljanja.



Slika 2.11: Metoda ranog zaustavljanja

Na grafiku sa slike 2.11 vidimo da je, na početku procesa treniranja, vrijednost funkcije gubitka na trening i validacionom skupu velika. Kako se treniranje nastavlja, vrijednost funkcije gubitka na trening skupu opada i postaje manja, što znači da se model poboljšava i da bolje predviđa vrijednosti na trening skupu. Međutim, postoji mogućnost da se model prenauči, tj. da se prilagodi trening skupu previše. To se obično manifestuje kroz rast vrijednosti funkcije gubitka na validacionom skupu. Cilj ranog prekida treniranja je da se zaustavi treniranje u trenutku kada model počne da se previše prilagođava podacima, tj. kada vrijednost funkcije gubitka na validacionom skupu počne da raste. To se može uočiti na grafiku 2.11 kao tačka u kojoj se linija koja prikazuje vrijednost funkcije gubitka na validacionom skupu počinje da raste, a linija koja prikazuje vrijednost funkcije gubitka na trening skupu nastavlja da pada (označeno isprekidanom linijom). Dakle, optimalan broj epoha za treniranje datog modela je upravo definisan tom tačkom. Treniranje modela

sa većim brojem epoha sigurno će dovesti do prenaučivosti. Uobičajeno je da se metoda ranog zaustavljanja kombinuje sa drugim metodama regularizacije, poput L2 ili metode *dropout*.

2.2.3 Optimizacioni algoritmi

Optimizacioni algoritmi su važan dio procesa treniranja modela mašinskog učenja, a njihov cilj je pronaći vrijednosti parametara modela koji minimizuju funkciju gubitka. Najjednostavniji način ažuriranja parametara u optimizacionom algoritmu je da se parametri mijenjaju u suprotnom smjeru od gradijenta funkcije gubitka. Gradijent funkcije gubitka pokazuje smjer povećanja funkcije, međutim, obzirom da u procesu optimizacije želimo da minimizujemo funkciju gubitka, onda se parametri mijenjaju u suprotnom smjeru od gradijenta. Algoritam se sastoji od iterativnog prilagođavanja težina modela u smjeru negativnog gradijenta funkcije gubitka. Korišćenjem izvoda funkcije, metod gradijentnog spuštavanja pronalazi pravac najvećeg pada funkcije u tački, i kreće se u tom smjeru dok se ne postigne konvergencija ili dok se ne dostigne maksimalan broj iteracija. Klasični metod gradijentnog spuštavanja koristi cjelokupni skup trening podataka kako bi ažurirao parametre modela što je memorijski zahtjevno i sporo. Postoje dvije varijante gradijentnog spuštavanja koje imaju bolje performanse nad velikim skupovima podataka i to su: *mini-batch* gradijentno spuštavanje i stohastično gradijentno spuštavanje (eng. *Stochastic Gradient Descent* - SGD). *Mini-batch* gradijentno spuštavanje računa gradijent funkcije gubitka na nasumično izabranom podskupu ukupnog skupa podataka, umjesto da računa gradijent funkcije gubitka na cijelom skupu podataka. Kada je veličina podskupa jednaka jednom uzorku onda govorimo o SGD algoritmu.

Prethodno pomenuti algoritmi imaju korak učenja koji je konstantan tokom čitavog procesa treniranja što može prouzrokovati oscilacije u blizini minimuma funkcije gubitka a samim tim i sporu konvergenciju. Takođe, postoji rizik da se algoritam zaustavi u lokalnom minimumu, a ne u globalnom minimumu funkcije gubitka. U praksi, se često koriste optimizacioni algoritmi sa varijabilnim korakom učenja koji se smanjuje tokom vremena kako bi se postigla bolja konvergencija. Optimizacioni algoritam *Adam* se često koristi u oblasti dubokog učenja. Štaviše, ovaj algoritam je predložen kao podrazumijevana metoda optimizacije u Stanfordovom kursu o dubokom učenju za kompjutersku viziju pod nazivom „CS231n: Konvolutivne neuronske mreže za vizuelno prepoznavanje“ [23].

2.3 Istorija i razvoj konvolucionih neuralnih mreža

Konvencionalne neuralne mreže se sastoje od vještačkih neurona koji obrađuju informacije linearno, odnosno vrše matematičku operaciju sabiranja težinskih koeficijenata vezanih za ulazne podataka i primjenu aktivacione funkcije na težinski sumu kako bi se dobio izlazni signal. Prvi model vještačkog neurona predložen je davne 1943. godine u radu [24], koji je nakon toga korišćen u raznim ANN mrežama, kao što su višeslojni perceptroni (eng. *Multi Layer Perceptron* - MLP). U odnosu na broj slojeva MLP mreža se smatra plitkom neuralnom mrežom, koj obično ima od 1 do 3 skrivena sloja između ulaznog i izlaznog sloja. Skriveni slojevi MLP arhitekture su većinom potpuno povezani (eng. *Fully Connected* - FC) slojevi, što znači da su svi neuroni u prethodnom sloju povezani sa svim neuronima u skrivenom sloju. Svaki sloj u MLP mreži se sastoji od više neurona koji obrađuju informacije na isti način - primjenom težina na ulazne podatke i primjenom aktivacione funkcije na težinski zbir kako bi se dobio izlazni signal. MLP modeli su i dalje popularni jer su jednostavni i efikasni. Takođe, ovi modeli su inspirirali razvoj drugih vrsta neuralnih mreža koje se danas koriste u mnogim oblastima.

Istraživački rad Hubela i Wiesel, koji je 1981. godine krunisan Nobelovim nagradom za fiziologiju ili medicinu, u velikoj mjeri je pomogao u razumijevanju kompleksnih procesa vizualne percepcije i procesuiranja vizuelnih informacija u mozgu. Njihovo istraživanje je pokazalo da je vizuelni korteks organizovan tako da se određene grupe neurona, specijalizovane za prepoznavanje određenih karakteristika vizuelnih podražaja, aktiviraju samo kada je prisutan određeni podražaj. Ovaj koncept hijerarhijskog procesiranja informacija u vizuelnom kortikalnom sistemu bio je ključan za razvoj konvolucionih neuralnih mreža. Kao što se u biološkom vizuelnom sistemu svaki neuron aktivira samo kada je stimuliran u ograničenom području vidnog polja, što se naziva receptivno polje, tako i u CNN mreži svaki neuron obrađuje podatke iz svog receptivnog polja.

Pretečom konvolucionih neuralnih mreža smatra se neuralna mreža koju su Fukushima i Miyake predložili 1982. godine u svom radu [25]. Predložena neuralna mreža pod nazivom *Neocognitron* je imala sposobnost razvrstavanja sličnih objekata u istu kategoriju, nezavisno od njihove pozicije i određenih morfoloških varijacija na način na koji to radi biološki vizuelni sistem. Međutim, pravi proboj u korištenju neuralnih mreža dogodio se 1989. godine kada je Yann LeCun formulisao algoritam propagacije unazad za učenje prvog konvolucionog modela neuralne mreže zvanog LeNet [26]. U poređenju s klasičnim pristupima mašinskog učenja, ovaj model je imao nenadmašnu preciznost kla-

sifikacije ručno pisanih brojeva iz MNIST skupa podataka [27]. Ipak, popularnost ovog metoda u tom periodu nije dugo trajala iz razloga što je bilo teško optimizovati duboke konvolucione mreže za male ili srednje skupove podataka. Osim toga, bilo je i računarski zahtjevno sa stanovišta dostupnih hardverskih resursa.

Jedan od najvažnijih događaja u oblasti kompjuterske vizije i mašinskog učenja predstavlja izazov koje se organizuje od 2010. godine pod nazivom *ImageNet Large Scale Visual Recognition Challenge* (ImageNet LSVRC). Ovaj izazov koji se odnosi na razvoj softvera za klasifikaciju i detekciju predmeta i scena iz slika postao je prekretnica za primjenu dubokih CNN mreža u oblasti kompjuterske vizije. Dana 30. septembra 2012. godine, Krizhevsky i saradnici su prvi put predložili duboki CNN model, takozvani AlexNet [28], koji je ujedno predstavljao i početak dubokog učenja. AlexNet je sinonim za CNN mrežu sa osam slojeva koja je klasifikovala 1,3 miliona slika visoke rezolucije u 1000 klasa sa greškom od 16,4 procenta, što je bilo oko 10 procenata manje od drugog najboljeg metoda koji koristi tradicionalni pristup mašinskog učenja. Ovaj revolucionarni rezultat, kao i rezultati dubokih CNN mreža u prethodnim godinama takmičenja, postignut je upotrebom grafičkih procesorskih jedinica (eng. *Graphics Processing Unit* - GPU) koje su omogućile brže treniranje dubokih CNN mreža na velikim skupovima podataka. Godine 2013, model nazvan ZFNet, poznat po imenima njegovih tvoraca Zeiler i Fergus, trijumfovao je na takmičenju ImageNet LSRVC. Arhitektura ZFNet je bila ista kao i AlexNet, dok su autori promijenili određene hiperparametre. Konkretno, u prvom konvolucionom sloju, veličina filtera je promenjena sa (11×11) sa korakom 4) na (7×7) sa korakom 2). Zatim je u konvolucionim slojevima 3, 4 i 5, broj filtera je povećan sa 384, 384, 256 na 512, 1024 i 512, respektivno. Zahvaljujući pažljivom izboru hiperparametara, zabilježeno je značajno smanjenje greške klasifikacije na 11,7 procenata. Na sljedećem takmičenju posebnu pažnju je privukao VGGNet model za klasifikaciju slika, osmišljen od strane tima *Visual Geometry Group* sa Univerziteta Oxford. Članovi VGG tima smatrali su da će se sa povećanjem dubine CNN mreže dobiti manja greška u klasifikaciji slika. Njihova CNN mreža sa 19 slojeva imala je grešku procjene od 7,3 procenta [29]. Godine 2014, Google je predstavio GoogLeNet duboku CNN arhitekturu sačinjenu od 22 sloja sa greškom procjene od 6,7 procenta [30]. Ono što je bilo karakteristično za ovu arhitekturu je da nije sadržala FC slojeve te je na taj način ukupan broj parametara mreže smanjen na 5 miliona, što je 12 puta manje od AlexNet arhitekture i oko 28 puta manje od VGG arhitekture. Pored toga, ključna inovacija u njihovom pristupu bila je lokalna jedinica sa paralelnim konvolucionim granama nazvana *inception modul*. Ovaj konvolu-

cioni modul se više puta pojavljivao u mreži umjesto konvolucionih slojeva i bio je temelj GoogleNet arhitekture. Godine 2015, grupa istraživača ispred Microsoft korporacije je predstavila novu ideju za poboljšanje performansi dubokih CNN modela, pod nazivom ResNet [31]. Glavna inovacija ResNet-a je upotreba residualnih blokova, koji se sastoje od nekoliko slojeva, a svaki blok ima skraćenu ili *skip* konekciju koja je omogućavala da se ulaz direktno doda na izlaz bloka. *Skip* konekcija pokazala se kao efikasan način rješavanja problema nestajućih gradijenata u veoma dubokim neuronskim mrežama. Ideja *skip* konekcija i residualnih blokova proširila na mnoge druge arhitekture koje su postigle veliki uspjeh u različitim aplikacijama u oblasti dubokog učenja.

Ključna prednost CNN modela u odnosu na konvencionalne neuralne mreže jeste njihova sposobnost da obavljaju ekstrakciju karakteristika i prepoznavanje slika. Tradicionalne metode mašinskog obično zahtijevaju predobradu podataka i ekspertsko znanje u oblasti primjene dok CNN mreže mogu automatski izvući bitne karakteristike iz sirovih podataka. Međutim, duboke CNN mreže imaju određena ograničenja, kao što je visoka računaska složenost, što zahtijeva specijalnu hardversku podršku za obuku, kao i veliku količinu podataka za treniranje kako bi postigle razumnu sposobnost generalizacije. Ovo je posebno izraženo kod 2D CNN mreža, gdje se računaska složenost povećava eksponencijalno sa dimenzijama slike i veličinom konvolucionog filtera. Konkretno, rezultat konvolucije slike dimenzija $N \times N$ i konvolucionog filtera dimenzija $K \times K$ će imati računsku složenost $O(N^2K^2)$ u 2D CNN mrežama. Sa druge strane, 1D CNN mreže sa signalom dužine N i filtrom veličine K imaju znatno manju računsku složenost $O(NK)$, koja se linearno povećava sa dužinom signala i veličinom filtera. Ovo znači da su 1D CNN mreže manje računski zahtjevne od 2D CNN mreža i mogu biti pogodne za obradu podataka u stvarnom vremenu i za niskobudžetne aplikacije.

Visok nivo performansi 1D CNN mreže postigle su u nekoliko oblasti kao što su medicina, mašinsko, građevinsko i avio inženjerstvo [32]. U svrhu detekcije i klasifikacije abnormalnih ponašanja u moždanim talasima pacijenata, pored ostalih razmatranih metoda u [33] korišćene su i 1D CNN mreže. Uzimajući u obzir parametarsku efikasnost korišćene CNN arhitekture, zaključeno je da se ista može implementirati u nosivom uređaju za detekciju artefakata u realnom vremenu. Autori rada [34] su koristili 1D CNN mreže za detekciju i identifikaciju aritmije u srčanim talasima i zaključili da pružaju brze i pouzdane dijagnoze u toj oblasti. Sistemi za nadgledanje stanja građevinskih i mašinskih struktura bazirani na vibracionim signalima omogućavaju detekciju oštećenja na pomenutim strukturama primjenom 1D CNN mreža. Jednodimenzionalne CNN mreže imaju

sposobnost automatskog odabira reprezentativnih karakteristika na osnovu „sirovog“ vibracionog signala, što omogućava njihovu primjenu na bilo koju građevinsku infrastrukturu [35]. CNN mreže se mogu uspješno koristiti i za procjenu štete nakon zemljotresa na osnovu podrhtavanja tla [36]. Naime, rezultati istraživanja pokazuju da je 1D CNN model u poređenju sa 2D CNN modelom, smanjio broj parametra za obuku za 86%, korišćenje memorije za 69%, vrijeme predobrade za 99,9%, vrijeme obuke za 96% i vrijeme testiranja za 90%.

Detekcija kvarova na motorima pomoću 1D CNN mreža obrađena je u radovima [37], [38] i [39]. Vibracioni signali detektovani pomoću senzora postavljenih na različitim djelovima motora korišćeni su za trening 1D CNN mreža. Pokazano je da 1D CNN mreže postižu veći nivo preciznosti u slučaju detekcije i klasifikacije kvarova u odnosu na druge metode. Takođe, eksperimentalni rezultati pokazuju da se pomoću 1D CNN mreža može vršiti praćenje stanja motora u stvarnom vremenu. Ipak, neki od nedostataka 2D CNN mreža, poput potrebe za velikom količinom podataka za obuku i složenost mreže, i dalje se prenose na 1D CNN mreže [40]. U cilju prevazilaženja ovog problema, autori rada [41] koristili su metode poput augmentacije podataka, normalizacije podataka i tehnike *dropout* za poboljšanje performansi 1D CNN mreža.

2.4 Karakteristike konvolucionih neuralnih mreža

Konvolucione neuralne mreže su dobile naziv po konvoluciji, jednoj od najvažnijih operacija u obradi signala i slike. U oblasti digitalne obrade slike operacija konvolucije, odnosno konvolucioni filtri, koristi se za izoštravanje i zamućivanje slike, detektovanje ivica, filtriranje šuma i još mnogo toga. Sa aspekta neuralnih mreža, operacija konvolucije predstavlja ključni alat koji se koristi u konvolucionim neuralnim mrežama za prepoznavanje objekata, klasifikaciju i segmentaciju slika. U odnosu na klasične neuralne mreže, primjena konvolucije u konvolucionim slojevima CNN mreža omogućava smanjenje broja parametara u mreži, smanjenje memorijskih zahtjeva i bolju efikasnost u obradi velikih skupova podataka. Na primjer, kada se slika dimenzija 300×300 piksela koristi kao ulazni podatak za klasičnu neuralnu mrežu, u ulaznom sloju treba rezervisati po jedan neuron za svaki piksel što je ukupno 90000 neurona. Za slike u boji, koje se predstavljaju sa tri kanala RGB boja, broj neurona u ulaznom sloju povećava se na 270000. Međutim, s obzirom na to da su susjedne vrijednosti signala ili piksela slike uglavnom u korelaciji i da se mogu izdvojiti određene karakteristike koje opisuju analizirane signale ili slike, kon-

volucione neuralne mreže upravo omogućavaju pronalaženje specifičnih karakteristika u podacima. Ovo se odnosi na koncept **rijetkih interakcija** (eng. *sparse interactions*) u CNN mrežama, gdje se mreža fokusira na lokalne informacije umjesto da uzima u obzir cjelokupne globalne informacije. Rijetke interakcije se postižu korišćenjem konvolucionog filtra (kernela) ili detektora karakteristika manjih dimenzija u odnosu na dimenzije ulaznog signala. Kerneli omogućavaju mreži da se fokusira samo na lokalne informacije u signalu, što znači da se karakteristike u signalu prepoznaju samo u malom području u kojem se nalaze, umjesto da se uzima u obzir cjelokupni signal. Tačnije, svaki neuron obrađuje samo djelove signala koji su unutar njegovog receptivnog polja koje je određeno veličinom kernela. U prvim slojevima CNN mreže, aktivacija neurona se događa kada je neuron podražen odgovarajućim ulaznim podacima odnosno kada se njegovo receptivno polje preklopi s određenim dijelom ulaznog signala, što upravo ukazuje na prepoznavanje neke karakteristike u ulaznom signalu. U narednim, dubljim slojevima, svaki neuron obrađuje podatke iz više prethodnih neurona i može prepoznati složenije karakteristike koje se sastoje od kombinacije pojedinačnih karakteristika. Na primjer, neuroni u prvim slojevima 2D CNN mreže mogu služiti za prepoznavanje linija i ivica da bi neuroni u dubljim slojevima mogli prepoznati konture ili teksture objekta.

Primjena različitih kernela u konvolucionim slojevima CNN omogućava ekstrakciju različitih karakteristika iz ulaznog signala i kreiranje **mape karakteristika** (eng. *feature map*). U principu, svaki kernel generiše jednu mapu karakteristika, koja je u suštini filtrirana verzija signala na koji se primjenjuje kernel. Mapa karakteristika će sadržati visoke vrijednosti tamo gdje je konvolucija između kernela i određenog segmenta signala najveća. Sa druge strane, niske vrijednosti u mapi karakteristika će se pojaviti tamo gdje se kernel i signal ne poklapaju, odnosno gdje je konvolucija između kernela i određenog segmenta signala manja ili čak jednaka nuli. Svaki neuron koji koristi isti kernel dijeli iste parametre, odnosno težinske koeficijente koji taj kernel čine. Dakle, u CNN mrežama učenje se svodi na pronalazak odgovarajućih kernela. **Dijeljenje težina** (eng. *weight sharing*), odnosno kernela, kroz konvolucione slojeve utiče na smanjenje broja parametara koje je potrebno naučiti, čime se smanjuje vrijeme obuke. Zbog dijeljenja težina, konvolucionim slojevima CNN mreža imaju svojstvo **ekvivarijantnosti prema translaciji** (eng. *translational equivariance*), što znači da će karakteristike signala biti detektovane bez obzira na njihovu poziciju u signalu. Kod klasičnih NN mreža, podaci se obrađuju pomoću slojeva potpuno povezanih neurona, koji ne uzimaju u obzir lokalnu strukturu podataka. To znači da se svaki uzorak u podacima tretira kao potpuno nezavisan i da se svaki neuron

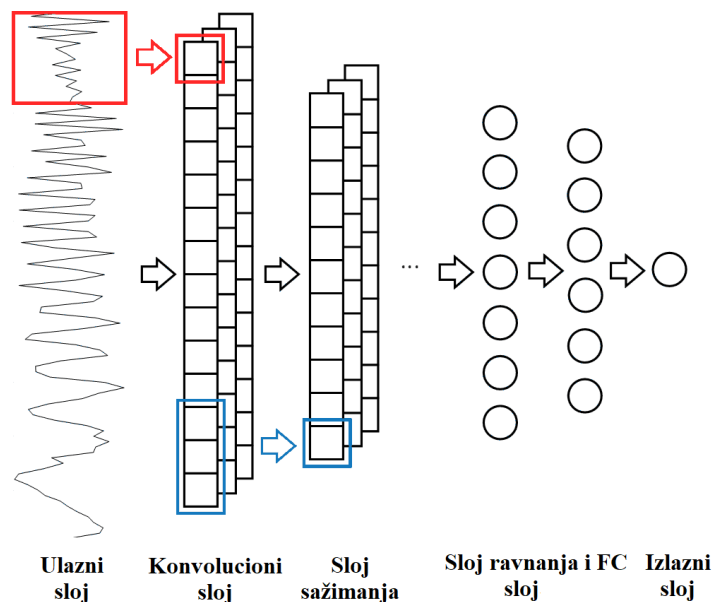
u mreži uči da reaguje na određene karakteristike u tom uzorku, što rezultira u velikom broju parametara koji se moraju naučiti. CNN mreže su posebno dizajnirane za rad sa podacima koji imaju lokalnu strukturu, što ih čini izuzetno efikasnim u prepoznavanju uzoraka u slikama, zvuku i drugim vrstama signala.

2.4.1 Arhitektura konvolucione neuralne mreže

Duboke konvolucione neuralne mreže sastoje se od velikog broja međusobno povezanih slojeva. Broj slojeva, vrsta slojeva i način na koji su slojevi povezani čini arhitekturu CNN mreže. Osnovni gradivni elementi CNN mreže su:

- Konvolucioni sloj,
- Sloj sažimanja (eng. *pooling layer*),
- Sloj ravnjanja (eng. *flatten layer*) i
- Potpuno povezani sloj.

Tipična CNN arhitektura sa gore navedenim slojevima je prikazana na slici 2.12.



Slika 2.12: Ilustracija tipične strukture 1D CNN modela

U ovom primjeru, jednodimenzionalni signal koji predstavlja zvuk vozila je doveden na ulaz u 1D CNN mrežu. Uobičajena praksa je da se konvolucioni i *pooling* slojevi sukcesivno dodaju u CNN arhitekturu. Nakon toga, obično slijedi sloj *flatten* i niz potpuno povezanih slojeva. Izlazni sloj sa slike 2.12 sadrži jedan neuron što znači da je ova mreža treba da predvidi jednu broječanu vrijednost koja predstavlja brzinu vozila.

Konvolucioni sloj

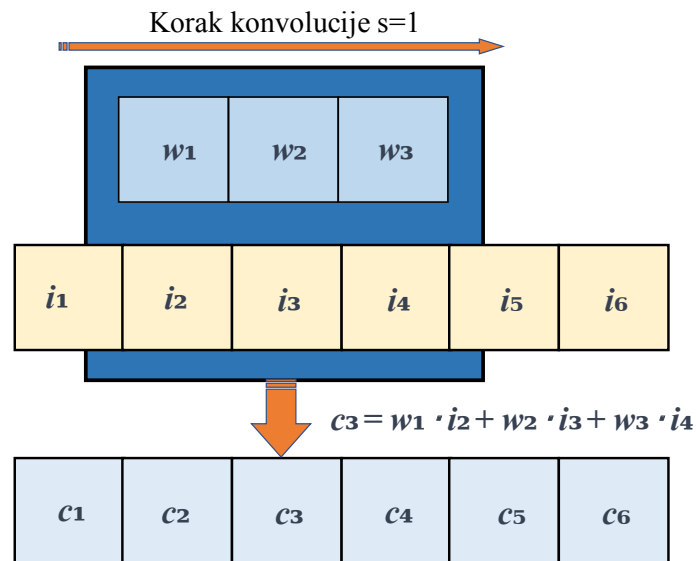
Konvolucioni sloj predstavlja osnovnu gradivnu jedinicu CNN mreže. U ovom sloju vrši se operacija konvolucije, koja podrazumijeva primjenu konvolucionog filtra na ulazni signal. Mogućnost korišćenja operacije konvolucije za detektovanje karakteristika u signalu može se objasniti primjenom teorije podudarajućih filtera (eng. *Matched Filter - MF*) [42]. Naime, izlazni signal iz MF se može zapisati kao konvolucija ulaznog signala $x(n)$ i impulsnog odziva filtra, $w(-n)$, odnosno:

$$y(n) = x(n) * w(-n) = \sum_m x(m)w(m-n) = \sum_m x(n+m)w(m) = x(n) *_c w(n), \quad (2.27)$$

gdje je $w(-n)$ vremenski obrnuta karakteristika koju tražimo u signalu. Dakle, detekciju prisustva određene karakteristike, $w(n)$, u signalu, $x(n)$, vrši se računanjem konvolucije signala $x(n)$ sa $w(-n)$.

Neka je jednodimenzionalni signal, dužine N , doveden na ulaz u CNN mrežu i neka se u konvolucionom sloju koristi kernel dužine $M \ll N$. Treba pomenuti da se u svakom konvolucionom sloju obično definiše više ovakvih kernela. U suštini, da bi detektovali K različitih karakteristika signala potrebno je primijeniti isto toliko filtera. Naime, konvolucioni filtri se pomijeraju po ulaznom signalu sa određenim korakom (eng. *stride*), množeći svaki element signala sa odgovarajućim elementom filtera i sabirajući dobijene vrijednosti, što daje jedan element izlaznog signala pri svakom pomjeraju. U opštem slučaju, primjenom kernela na ulazni signala dobija se rezultatni signal dužine $N - M + 1$. Ukoliko se gore opisana procedura pomijeranja kernela po ulaznom signalu ponovi za K različitih kernela, na izlazu iz konvolucionog sloja dobićemo mapu karakteristika dimenzija $K \times (N - M + 1)$. Dakle, ukupan broj parametara koje mreža treba da nauči jednak je broju filtara pomnoženim sa dužinom filtra, odnosno $K \times M$. Međutim, kao i kod standardnih potpuno povezanih neuralnih mreža, u konvolucionim slojevima se obično koristi i *bias* član što rezultira sa ukupno $K \times (M + 1)$ parametara u jednom konvolucionom sloju.

Na slici 2.13 je prikazan proces pomijaranja kernela dužine $M = 3$ po jednodimenzionalnom signalu. Pomijeranje se vrši korakom veličine 1, sve dok se ne obuhvate svi elementi ulaznog signala. Sa svakim korakom, računa se skalarni proizvod između težinskih koficijenata kernela: w_1 , w_2 i w_3 i djelova ulaznog signala koji se nalaze ispod kernela. U primjeru sa slike 2.13 to su: i_2 , i_3 i i_4 . Potom se skalarni proizvodi sabiraju kako bi se generisao izlazni element c_3 . Pomijeranje kernela se vrši dok se ne stigne do kraja signala.



Slika 2.13: Primjena konvolucionog filtra na 1D signal

U slučaju potpuno povezane neuralne mreže, svaki ulazni neuron se povezuje sa svakim izlaznim neuronom, što dovodi do velikog broja parametara koji se moraju naučiti. Konkretno, broj veza između slojeva je jednak proizvodu broja ulaznih neurona i broja izlaznih neurona, odnosno postoji $N \times K$ veza. Takođe, potpuno povezana neuralna mreža se može shvatiti kao poseban slučaj CNN mreže gdje je veličina kernela jednaka dužini ulaznog signala. Postojanje rijetkih konekcije kod CNN mreža nagovještava da su CNN mreže manje osjetljive na prenaučenosť i efikasnije sa stanovišta vremena trajanja obuke u odnosu na klasične neuralne mreže.

U CNN mrežama, često se koristi tehnika *padding* koja se sastoji od dodavanja određenih brojnih vrijednosti na početku ili kraju ulaznog signala kako bi izlazni signal konvolucije imao iste dimenzije kao ulazni signal. Tehnika *padding* koja zadovoljava ovo svojstvo često se naziva *same padding*. Opcija *full padding* proširuje mapu karakteristika za $M - 1$ po svakoj dimenziji, dok *valid padding* ne primjenjuje nikakav *padding* i kao takav smanjuje dimenzije mape karakteristika za isti isnos [43]. Ovo je posebno izraženo

u dubokim CNN mrežama gdje se sa svakim konvolucionim slojem smanjuju dimenzije mape karakteristika, ukoliko se ne primjenjuje tehnika *padding*. Osim očuvanja dimenzija mapa karakteristika kroz slojeve, opravdanje za korišćenje ove tehnike nalazi se i u činjenici da se početni i krajnji djelovi signala obrađuju manje puta od onih u sredini signala, što može dovesti do gubitka informacija na ivicama signala. Postoje različiti načini primjene tehnike *padding* u CNN mrežama, ipak najčešća tehnika je *zero-padding* koja uključuje dodavanje nula na ivice ulaznog signala. Ova tehnika se često koristi jer je jednostavna za implementaciju i ne dodaje dodatne informacije u ulazni signal.

Uz primjenu *padding*-a, pomjeraj kernela u odnosu na ulazni signal takođe ima značajan uticaj na dimenzije izlaznog signala. Veće vrijednosti koraka pomijeranja dovode do manjih izlaznih dimenzija, što može smanjiti rezoluciju izlaza i uticati na sposobnost mreže da prepozna detalje. Međutim, ovo može biti korisno u nekim slučajevima, npr. kada se želi smanjiti broj parametara u mreži ili uštedjeti na računarskim resursima potrebnim za obradu ulaznih signala. U zavisnosti od prirode signala vrši se i odabir ovog koraka. Obično se vrše uzastopni pomjeraji kernela, odnosno uzima se korak jednak jedinici. Međutim, u nekim slučajevima se može preskočiti jedan trenutak vremena u izračunavanju konvolucije, što bi značilo da se uzima koraka čija je vrijednost dva.

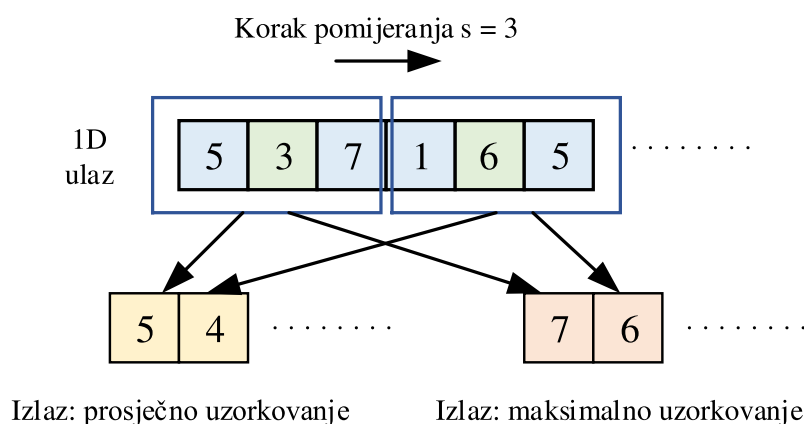
U opštem slučaju, za signal dužine N , kernel dužine M i korak konvolucije postavljen na vrijednost s , dužina izlaznog signala će biti $(N - M)/s + 1$. Dakle, vidimo da je broj uzoraka u izlaznom signalu smanjen za faktor s . U digitalnoj obradi signala, ova tehnika se naziva *downsampling* i označava smanjenje rezolucije digitalnog signala. Konačan izlaz iz konvolucionog sloja dobija se primjenom odgovarajuće aktivacione funkcije na mapu karakteristika. Obzirom da je konvolucija linearna operacija, aktivaciona funkcija je neophodna da bi dodala nelinearnost u neuralnu mrežu. Nakon primjene aktivacione funkcije, nova mapa karakteristika se koristi kao ulaz u sledeći sloj neuralne mreže. U CNN mrežama se najčešće koristi ReLU aktivaciona funkcija.

Zaključujemo da konvolucionni sloj u CNN mrežama ima nekoliko hiperparametara koji se mogu prilagođavati tokom postupka učenja i to su: broj kernela, veličina kernela, korak pomijeranja, *padding* i aktivaciona funkcija. Odabir ovih hiperparametara zavisi od prirode ulaznih signala, složenosti zadatka i namjene mreže. Prilagođavanje ovih hiperparametara je važan dio postupka učenja u CNN mrežama i u velikoj mjeri utiče na performanse modela.

Sloj sažimanja

Sloj sažimanja predstavlja važan sloj u CNN mrežama, a njegova primarna funkcija je smanjenje dimenzija mapa karakteristika. Primjenom sloja sažimanja se postiže smanjenje broja parametara koje je potrebno naučiti, poboljšava se računaska efikasnost i može se spriječiti pretreniranost modela. Slično kao konvolucionni sloj, u sloju sažimanja postoji kernel koji prolazi kroz signal, ali je razlika u tome što ovaj kernel nema nikakve težine. Umjesto konvolucije, ovaj kernel primjenjuje funkciju agregacije na vrijednosti unutar svog receptivnog polja, odnosno na dio mape karakteristika koji se obrađuje u datom koraku.

Sažimanje se obično primjenjuje nakon konvolucionog sloja i uključuje podjelu mapa karakteristika na određene regione, koje se nazivaju regioni sažimanja. Veličina svakog regiona sažimanja definisana je veličinom kernela. Obično se uzimaju najbliži nepreklopujući regioni iz mape karakteristika, iako je moguće definisati korak pomijeranja kernela na isti način kao i u konvolucionom sloju. Na svaki region sažimanja se primjenjuje određena funkcija sažimanja kako bi se dobila jedna izlazna vrijednost po regionu. Najčešće korišćena tehnika sažimanja je tehnika maksimalnog uzorkovanja (eng. *max pooling*), koja uzima najveću vrijednost kao reprezentativnu vrijednost određene regije. Uz maksimalnu vrijednost, postoji nekoliko drugih operacija koje se mogu primijeniti u sloju sažimanja, kao što su srednja (prosječna) vrijednost i L2 norma. Na slici 2.14 je prikazan proces pomijeranja prozora dužine $M = 3$ duž jednodimenzionalnog signala. Pomijeranje se vrši korakom $s = 3$. Takođe su prikazane dvije najčešće korišćene operacije sažimanja na primjeru jednodimenzionalnog signala.



Slika 2.14: Funkcije sažimanja: prosječno i maksimalno uzorkovanje

U mnogim situacijama, tehnika prosječnog uzorkovanja (eng. *average pooling*) se

pokazala boljom u odnosu na tehniku maksimalnog uzorkovanja zato što koristi sve vrijednosti unutar uzorkovane regije za izračunavanje prosjeka, za razliku od maksimalnog uzorkovanja koje koristi samo najveću vrijednost. Stoga, u određenim situacijama, prosječno uzorkovanje može biti bolja opcija od maksimalnog uzorkovanja obzirom da može dovesti do manje varijacije u procijenjenoj vrijednosti [39]. Ovo se posebno odnosi na podatke u kojima je prisutan šum, gdje se korištenjem prosječne vrijednosti unutar uzorkovane regije može smanjiti uticaj šuma.

Zahvaljujući sloju sažimanja, CNN mreže posjeduju svojstvo invarijantnosti na translaciju (eng. *translational invariance*). Translaciona invarijantnost se odnosi na situaciju u kojoj promjena položaja bitnih karakteristika u signalu ne dovodi do promjene mape karakteristika poslije sloja sažimanja. Naime, funkcija uzorkovanja će dovesti do toga da karakteristike koje se nalaze na različitim mjestima u signalu, budu smještene na otprilike istu lokaciju nakon dovoljnog broja ponavljanja funkcije uzorkovanja, odnosno nakon više slojeva sažimanja. Međutim, treba obratiti pažnju na veličinu prozora uzorkovanja jer je utvrđeno da postoji veza između generalizacije modela i translacione invarijantnosti. Povećanje veličine prozora za uzorkovanje dovodi do poboljšane invarijantnosti, ali smanjuje sposobnost generalizacije modela [44].

Potpuno povezani sloj

Nakon sloja sažimanja, dobijeni izlazni podaci mogu se koristiti kao ulazni podaci za sledeći konvolucionni sloj, ukoliko postoji više konvolucionih slojeva u mreži. Takođe, ovi izlazni podaci se mogu izravnati (eng. *flatten*) i koristiti kao ulazni podaci za potpuno povezani sloj neuralne mreže, gdje se dalje obrađuju kako bi se dobili izlazni rezultati. Sloj *flatten* omogućava transformaciju višedimenzionalne mape karakteristika koju je sloj sažimanja proizveo u jednodimenzionalni oblik.

Izlazni vektor iz sloja *flatten* predstavlja ulazni sloj potpuno povezane neuralne mreže. Ovaj sloj se takođe naziva i gusto povezani sloj (eng. *dense layer*) jer je svaki neuron u sloju gusto povezan sa svim neuronima iz prethodnog sloja. Svrha potpuno povezanog sloja u CNN mreži je detektovanje određenih karakteristika u signalu. Preciznije, svaki neuron u potpuno povezanom sloju odgovara određenoj karakteristici koja može biti prisutna u signalu. Vrijednost koju neuron prenosi na sljedeći sloj predstavlja vjerovatnoću da se ta karakteristika nalazi u signalu.

Glava 3

Procjena brzine i klase brzine vozila

U ovoj sekciji je predstavljen 1D CNN model koji je korišćen za procjenu brzine vozila na osnovu zvučnih zapisa koje vozilo generiše prilikom prolaska pored akustičnog senzora, odnosno mikrofona. U eksperimentalnom pristupu, analiziran je uticaj različitih hiperparametara modela, kao što su broj slojeva, broj i veličina konvolucionih filtera, dužina trajanja trening procesa i optimizaciona funkcija, na evaluaciju brzine vozila. Cilj je bio pronalaženje optimalne arhitekture modela koja bi pružila najbolje rezultate u procjeni brzine vozila. Tabela 3.1 prikazuje arhitekturu predloženog 1D CNN modela za procjenu brzine vozila na osnovu zvučnih zapisa. Ova arhitektura detaljno opisuje strukturu i konfiguraciju slojeva, kao i brojne druge hiperparametre koji su korišćeni u modelu.

Predloženi 1D CNN model za procjenu brzine vozila sastoji se od ukupno 8 konvolucionih slojeva. Prvi konvolucioni sloj ima 32 kernela veličine 32, što mu omogućava da izdvoji dominantne karakteristike iz zvučnih zapisa vozila i suzbije šum visoke frekvencije [40]. U nižim slojevima, veličina kernela i korak su smanjeni, a za bolju reprezentaciju karakteristika primijenjeni su uzastopni mali kerneli veličine 2. Nakon dva uzastopna konvoluciona sloja, koristi se sloj sa prosječnim sažimanjem radi smanjenja veličine mape karakteristika i određivanja prostorno nezavisnih karakteristika signala. Nakon pretvaranja dobijenog konvolucionog izlaza u jednodimenzionalni oblik, slijede dva potpuno povezana sloja. Prvi potpuno povezani sloj ima 100 neurona, dok drugi ima 10 neurona. Izlaz mreže je jedan neuron koji daje procjenu brzine vozila u kilometrima na sat (km/h).

Svi konvolucioni slojevi u modelu koriste ReLU aktivacionu funkciju, koja omogućava modelu da nauči nelinearne veze između ulaza i izlaza. Međutim, na potpuno povezanim slojevima se ne primjenjuje aktivacija, odnosno izlaz iz mreže je linearna kombinacija

Tabela 3.1: Predloženi 1D CNN model

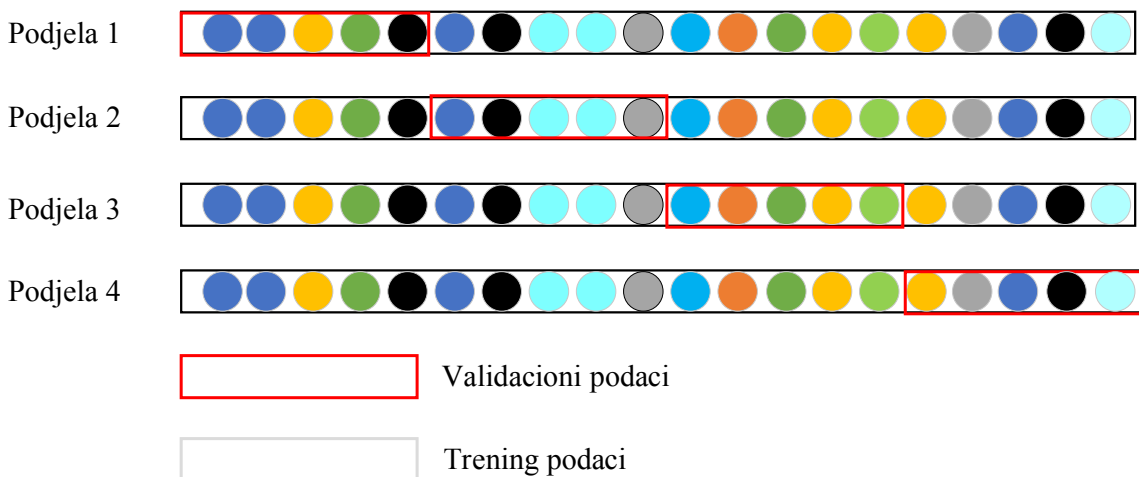
Sloj	# kernela	Veličina kernela	Pooling faktor	Stride	Dimenzije izlaza
Input	–	–	–	–	264600
Conv1D	32	32	–	16	32×16536
Conv1D	32	8	–	4	32×4133
AveragePool	–	–	2	2	32×2066
Conv1D	64	2	–	2	64×1033
Conv1D	64	2	–	2	64×516
AveragePool	–	–	2	2	64×258
Conv1D	96	2	–	2	96×129
Conv1D	96	2	–	2	96×64
AveragePool	–	–	2	2	96×32
Conv1D	128	2	–	2	128×16
Conv1D	128	2	–	2	128×8
Flatten	–	–	–	–	1024
Dense	–	–	–	–	100
Dense	–	–	–	–	10
Output	–	–	–	–	1

ulaza. Ova arhitektura modela je pažljivo odabrana kako bi se postigla što bolja procjena brzine vozila na osnovu zvučnih zapisa. Kroz eksperimente i evaluaciju, utvrđeno je da ova arhitektura daje zadovoljavajuće rezultate i pokazuje sposobnost modela da nauči i reprezentuje važne karakteristike zvučnih signala vezanih za brzinu vozila.

Procjena brzine je posmatrana kao regresioni i kao klasifikacioni problem. Kod regresije, izlaz modela je procijenjena vrijednost brzine, a mjera tačnosti je kvadratni korijen srednje kvadratne greške procjene brzine. Kod klasifikacije, vrši se diskretizacija opsega brzina, od najmanje do najveće iz skupa mjerenja, i procjenjuje se kojem opsegu pripada predmetna brzina. Tačnost klasifikacije brzine je prikazana kao vjerovatnoća pravilne procjene klase kojoj brzina pripada.

3.1 Unakrsna validacija

Obzirom da je veličina skupa podataka kreiranog za potrebe ovog istraživanja relativno mala, odabir najpogodnijeg modela izvršen je pomoću iterativne unakrsne validacione metode (eng. *iterated K-fold Cross Validation* - CV). Ovaj metod podrazumijeva treniranje modela u više iteracija sa različitom raspodjelom podataka u svakoj iteraciji. Naime, u toku jedne iteracije, cjelokupan skup podataka se dijeli na K jednakih skupova (obično 5 ili 10). Zatim se model trenira i testira K puta, gdje se u svakoj podjeli podataka jedan od K skupova koristi kao validacioni skup, a preostalih $K - 1$ skupova kao trening skup. Na kraju, prosječna uspješnost modela se računa na nivou svih izvršenih iteracija u cilju smanjenja oscilacija performansi i obezbjeđivanja bolje generalizacije modela. Na slici 3.1 prikazana je podjela podataka primjenom unakrsne validacije, za $K = 4$.



Slika 3.1: Prikaz podjele podataka primjenom CV metode

Konkretno, u problemu procjene brzine vozila, prije svake CV procedure, skup podataka koji je vezan za određeno vozilo se zadržava za testiranje, dok se podaci od preostalih 12 vozila koristi za obuku i validaciju modela. Potom se podaci vezani za tih 12 vozila dijele na $K = 4$ podskupa kroz ukupno $M = 10$ iteracija, što znači da se ukupno odradi $K \times M = 40$ odvojenih obuka modela. Prilikom svake podjele podataka, jedan od podskupova se zadrži za testiranje, dok se ostala tri podskupa koriste za obuku modela. Nakon toga, model se testira na zadržanom podskupu i bilježi se njegova performansa. Postupak se ponavlja dok se ne testira svaki od četiri podskupa. Važno je napomenuti da se prije svake podjele podaci promiješaju kako bi se osiguralo da se model ne prilagođava određenom redosljedu podataka.

Postoji više vrsta unakrsne validacije. U ovom radu je korišćena klasifikovana (eng. *stratified*) unakrsna validacija iz razloga što se skup podataka sastoji od više klasa brzina. Upotreba ovog metoda osigurava da svaki podskup ima približno jednaku raspodjelu klasa, odnosno da se svaka brzina nađe u svakom podskupu u približno jednakoj razmjeri, čime se smanjuje uticaj neravnomjerne raspodjele uzoraka u klasama na performanse modela.

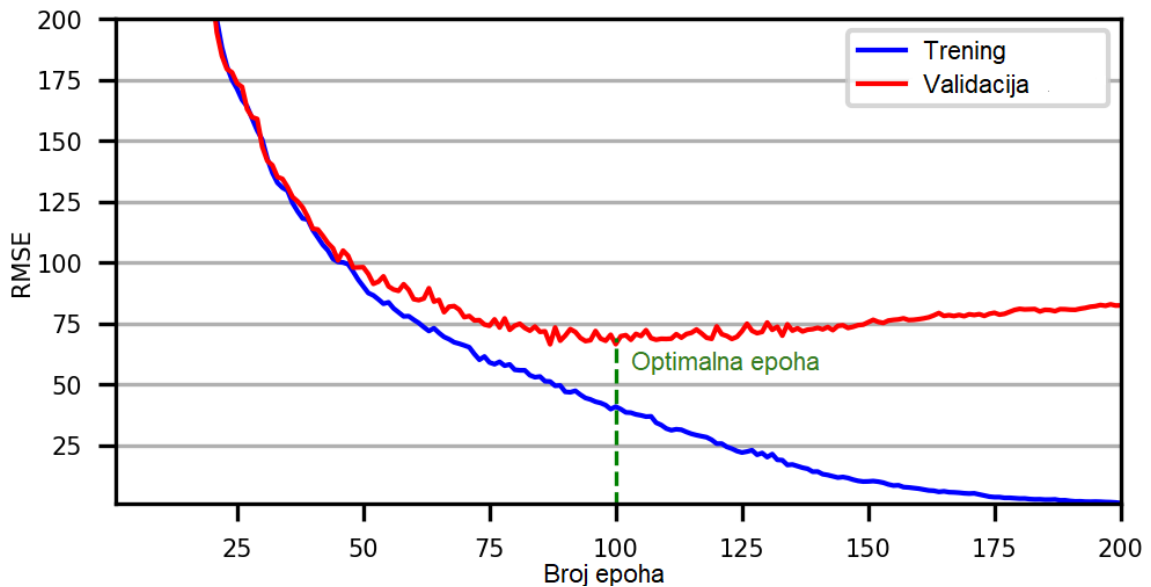
3.2 Trening procedure

Nakon podjele podataka primjenom odgovarajuće unakrsne validacije, vrši se treniranje i validacija predloženog 1D CNN modela na odgovarajućem dijelu podataka. U ovom radu testirane su i ocijenjene dvije procedure obuke predloženog 1D CNN modela za procjenu brzine vozila na osnovu zvučnog signala, jedna zasnovana na određivanju optimalnog broja epoha obuke, a druga na pamćenju parametara modela koji postiže minimalnu funkciju gubitka na validacionom skupu podataka.

3.2.1 Pristup sa optimalnom epohom

Prvi pristup bazira se na bilježenju vrijednosti funkcije gubitka za trening i validacioni skup podataka nakon svakog prolaza kroz podatke, odnosno nakon svake epohe, u svakoj iteraciji. Na ovaj način omogućeno je praćenje napredovanja modela kroz proces obuke. U suštini, cilj je dobiti srednju vrijednost funkcije gubitka preko svih iteracija i svih *k-fold* podijela. Ovaj pristup omogućava praćenje performansi modela na osnovu funkcije gubitka i daje smjernice za dalje korake u obuci. Praćenje srednje vrijednosti funkcije gubitka tokom epoha omogućava uvid u trend smanjenja ili povećanja gubitka tokom vremena. To može biti korisno za odabir optimalne epohe zaustavljanja treninga ili za identifikaciju potencijalnih problema u prilagođavanju modela. Na slici 3.2 prikazane su srednje vrijednosti funkcije gubitka za trening i validacione podatke jednog vozila iz skupa podataka (Kia Sportage). Slični trendovi u promjeni vrijednosti funkcije gubitka dobijaju se i za ostala vozila.

Grafik predstavljen na slici 3.2 se može koristiti za evaluaciju performansi modela na određenom vozilu. Na x-osi se nalazi broj epoha, odnosno prolaza kroz trening podatke, a na y-osi se nalazi vrijednost funkcije gubitka. Plava linija predstavlja srednju vrijed-



Slika 3.2: Funkcija gubitka za trening i validaciju vozila Kia Sportage

nost funkcije gubitka za trening podatke, dok crvena linija predstavlja srednju vrijednost funkcije gubitka za validacione podatke. Kako se broj epoha povećava, vrijednost funkcije gubitka se smanjuje, što znači da model sve bolje predviđa brzinu vozila na osnovu zvučnog signala. Međutim, može se primijetiti da se, u određenoj fazi obuke, vrijednost funkcije gubitka za validacione podatke počinje povećavati, dok se vrijednost funkcije gubitka za trening podatke nastavlja smanjivati. To ukazuje na problem pretreniranost modela, odnosno da model postaje previše specifičan za trening podatke i da ne generalizuje dobro na novim podacima. Na grafiku je takođe označena (isprekidanom zelenom linijom) optimalna epoha, odnosno epoha u kojoj se postiže najmanja vrijednost funkcije gubitka za validacione podatke. Optimalna epoha omogućava određivanje trenutka u kojem treba prekinuti obuku modela jer dalje obučavanje modela neće značajno poboljšati njegove performanse već dovodi do pretreniranosti.

Ovaj pristup, koji se fokusira na praćenje ukupnog napretka modela tokom obuke, omogućava nam da steknemo širu sliku o ponašanju modela tokom vremena. Generalno, postoje tri ključna aspekta koje možemo analizirati u odnosu na krive učenja (eng. *learning curve*) sa slike 3.2 i to su: brzina konvergencije, stabilnost modela i trend funkcije gubitka. Brzina konvergencije nam govori koliko brzo model nauči iz podataka tokom treninga. Ukoliko funkcija gubitka brzo opada tokom iteracija, to ukazuje na efikasno učenje modela i sposobnost prilagođavanja podacima. S druge strane, sporija konvergencija može ukazivati na složenost problema ili potrebu za prilagođavanjem hiperparametara

kako bi se ubrzao proces učenja [45]. Prateći trendove gubitka možemo utvrditi stabilnost modela, odnosno da li gubitak opada ravnomjerno, ima oscilacija ili se stabilizuje na određenoj vrijednosti. Stabilnost modela je važan faktor za pouzdanost i dosljednost performansi. Funkcija gubitka koja konzistentno opada tokom iteracija ukazuje na stabilnost modela. Ovo je poželjno jer znači da model dosljedno poboljšava svoje performanse tokom procesa obuke. Međutim, velike fluktuacije u funkciji gubitka, mogu ukazivati na nestabilnost ili preosjetljivost modela na male promjene u podacima. U takvim slučajevima, potrebno je izvršiti prilagođavanje modela kako bi se postigla veća stabilnost.

3.2.2 Pristup sa minimalnim gubitkom na validacionom skupu

Drugi pristup obučavanja predložene 1D CNN mreže fokusira se na pronalaženje modela sa optimalnim parametrima u smislu minimalnog gubitka na validacionom skupu. To znači da se tokom svake obuke modela (od ukupno 40 obuka) prati gubitak na validacionom skupu kako bi se sačuvali parametri modela koji postiže minimalan gubitak na validacionom skupu. Dobijeni model se zatim upoređuje sa trenutno najboljim modelom u smislu funkcije gubitka na validacionom skupu. Ako se isti pokaže bolji od trenutno najboljeg modela, postaje trenutno najbolji model. U suprotnom, odbacuje se. Ova procedura daje model sa minimalnim gubitkom na validacionom skupu nakon 40 obuka sa različitim kombinacijama podataka koje su korišćene za validaciju.

Čuvanje modela sa najmanjim gubitkom na validacionom skupu omogućava upotrebu tog modela za procjenu brzine vozila na novim podacima. Dakle, čuvaju se parametri modela koji je postigao najmanju grešku validacije u svim iteracijama obuke. Potom, za potrebe procjene brzine vozila na osnovu novog zvučnog signala, vrši se učitavanje parametara sačuvanog modela i procjena brzine vozila. Odabirom modela sa minimalnim gubitkom na validacionom skupu očekuju se bolje performanse na stvarnim podacima.

Glava 4

Eksperimentalni rezultati

U ovom poglavlju će biti prikazan korišteni skup anotiranih zvučnih zapisa, kao i rezultati procjene brzine vozila na tom skupu. Prvo će biti analizirani rezultati regresije, gdje je cilj procijeniti tačnu vrijednost brzine vozila na osnovu zvučnog zapisa. U kontekstu procjene brzine vozila, RMSE će se koristiti kao metrika za određivanje preciznosti modela u odnosu na stvarne brzine vozila. Potom će biti prikazani rezultati klasifikacije koja omogućava kategorizaciju vozila prema rasponu brzina na osnovu zvučnog signala. Prikazani rezultati će pokazati koliko uspješno model može klasifikovati brzine vozila u odgovarajuće klase.

Nastavak poglavlja će obuhvatiti raspravu o postignutim rezultatima, njihovoj pouzdanosti i eventualnim ograničenjima. Analiza rezultata će pružiti uvid u performanse modela u procjeni brzine vozila na osnovu zvučnih zapisa, kao i njihovu primjenjivost u praktičnim scenarijima.

4.1 Skup podataka

Istraživanje procjene brzine vozila koristeći audio-vizuelne podatke suočava se s izazovima nedostatka dostupnih skupova podataka u odnosu na druge istraživačke domene. To znači da je broj skupova podataka za takvu vrstu istraživanja vrlo ograničen. Kako bismo nadomjestili taj nedostatak, sakupili smo skup podataka koji se sastoji od audio-vizuelih snimaka vozila na putu koja prolaze pored kamere sa konstantnim brzinama. Svaki snimak u skupu podataka sadrži jednu vožnju jednog vozila. Korišćeno je trinaest

različitih vozila, sa ukupno 400 označenih audio-video snimaka. Skup podataka, poznat kao VS13 (Vehicle Speed 13 vehicles), u potpunosti je dostupan i kreiran je kako bi podstaknuo istraživače da razvijaju metode za procjenu brzine vozila bazi na audio-vizualnih podataka. VS13 je kreiran poštujući sledeće zahtjeve:

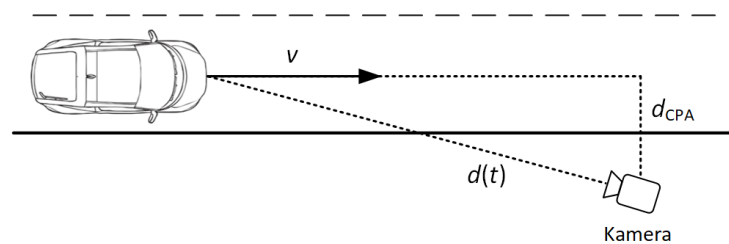
1. Svaki snimak sadrži jednu vožnju jednog vozila.
2. Snimci su napravljeni u urbanom okruženju.
3. Snimci su autentični terenski snimci.
4. Vozila su raznovrsna u pogledu proizvođača, godine proizvodnje, tipa motora (benzin ili dizel), snage i prenosa (ručni ili automatski).
5. Vozila su opremljena tempomatom (sistemom za kontrolu brzine), tako da brzina može biti održavana stabilnom dok vozilo prolazi pored kamere.

Prvi zahtjev podrazumijeva da je ovaj skup podataka fokusiran na procenu brzine pojedinačnih vozila umjesto na mjerenje prosječne brzine svih vozila na određenom dijelu puta, što je poznato kao brzina saobraćaja. Ova specifičnost omogućava detaljniju analizu brzine svakog vozila ponaosob, što je korisno u mnogim situacijama kao što su praćenje individualnih vozila, upravljanje saobraćajem ili analiza ponašanja vozača. Zahtjevi 2 i 3 su posebno važni u kontekstu akustične procjene brzine vozila. Kada se snimaju vozila u realnom okruženju, zvuk prolaska vozila (na osnovu kojeg se procjenjuje brzina) može biti narušen zvukom drugih vozila koja se nalaze u blizini ili prirodnim zvucima kao što su vjetar, cvrkut ptica i tome slično. Ovo predstavlja izazov za tačnu procjenu brzine vozila na osnovu zvuka, jer je potrebno izolovati ili filtrirati relevantni zvuk iz mješavine zvukova kako bi se dobile pouzdane informacije o brzini vozila. Ovaj skup podataka, sa svojim realnim terenskim snimcima, omogućava istraživačima da se uhvate u koštac sa tim izazovom i razviju napredne metode za akustičnu procjenu brzine vozila u realnom svijetu.

4.1.1 Prikupljanje podataka

Skup podataka je snimljen na lokalnom putu dužine 622 metara, udaljenom 90 metara od glavnog puta koji povezuje dva crnogorska grada: Podgoricu i Petrovac. Odabrani put

ima nekoliko ključnih prednosti. Prvo, njegova dužina omogućava vozačima da postignu stabilne brzine prije prolaska pored kamere za mjerenje brzine. Drugo, put je dovoljno izolovan od drugih saobraćajnica i izvora smetnji, što omogućava pouzdana mjerenja brzine bez većih ometanja. Treće, blizina drugih puteva omogućava ispunjenje zahtjeva 2 i 3, što osigurava raznovrsnost i reprezentativnost terenskih snimaka. Ovi autentični terenski snimci su od ključne važnosti za simuliranje realnih uslova vožnje.



Slika 4.1: Gore: Snimak postavke kamere. Dolje: Vozilo se kreće konstantnom brzinom v . Udaljenost između vozila i kamere u trenutku t , označena je sa $d(t)$, dok je d_{CPA} udaljenost na najbližoj tački susreta (eng. *Closest Point of Approach* - CPA).

Za snimanje skupa podataka, kamera *GoPro Hero5 Session* je korišćena kao glavni uređaj. Kamera je bila postavljena na stativu pored puta, na udaljenosti od otprilike 0,5 metara od same saobraćajnice, a na visini od oko 1,2 metra. Na slici 4.1 može se vidjeti snimak postavke kamere. Proces snimanja sesija se odvijao u vremenskom periodu od decembra 2019. do februara 2022. godine. Kako bi se prikupili raznovrsni podaci, ukupno trinaest vozila je korišćeno za snimanje. Tabela 4.1 prikazuje podatke o različitim vozilima i njihovim brzinama. Svaki red tabele predstavlja jedno vozilo, prva kolona sadrži naziv vozila (skraćeni naziv), dok druga kolona prikazuje brzine koje su zabilježene za to vozilo.

Tabela 4.1: VS13 vozila i brzine

Kratko ime vozila	Brzine (km/h)
Citroen C4 Picasso	35, 38, 41, 44, 48, 51, 54, 57, 59, 63, 65, 68, 72, 74, 78, 80, 83, 85, 87, 92, 94, 96, 101
Kia Sportage	31, 33, 35, 38, 41, 44, 46, 48, 51, 53, 55, 58, 61, 63, 65, 68, 69, 72, 74, 77, 78, 80, 83, 85, 86, 89, 91, 93, 96, 98, 100, 103, 105
Mazda 3	30, 33, 35, 38, 40, 43, 45, 47, 50, 52, 55, 57, 60, 62, 64, 67, 70, 72, 75, 79, 81, 84, 86, 88, 90, 92, 94, 96, 99, 101, 103, 105
Mercedes AMG 550	30, 33, 35, 38, 40, 42, 45, 47, 50, 52, 55, 58, 60, 62, 65, 67, 70, 73, 75, 78, 80, 82, 85, 87, 90, 93, 95, 98, 100, 105
Mercedes GLA	30, 33, 36, 39, 41, 42, 45, 47, 48, 49, 52, 54, 55, 59, 61, 63, 65, 68, 70, 72, 75, 78, 81, 83, 85, 88, 90, 92, 93, 96, 100, 101, 103, 104
Nissan Qashqai	35, 38, 40, 42, 45, 48, 50, 53, 55, 58, 60, 61, 64, 65, 68, 70, 73, 75, 78, 80, 82, 85, 88, 90, 93, 94, 96, 98, 102
Opel Insignia	31, 35, 38, 41, 44, 47, 50, 53, 55, 58, 61, 64, 66, 68, 70, 72, 73, 76, 78, 80, 83, 86, 89, 91, 94, 97, 100
Peugeot 208	30, 32, 34, 37, 40, 43, 45, 47, 50, 51, 54, 57, 60, 62, 64, 67, 68, 71, 73, 76, 77, 79, 82, 84, 87, 90, 92, 95, 96
Peugeot 3008	40, 43, 45, 47, 50, 52, 54, 55, 56, 58, 60, 61, 63, 65, 67, 68, 70, 72, 74, 75, 78, 80, 83, 85, 87, 89, 90, 92, 95, 97, 100
Peugeot 307	30, 33, 35, 38, 40, 43, 45, 47, 48, 50, 53, 56, 59, 60, 63, 66, 69, 72, 73, 76, 79, 82, 85, 88, 91, 94, 97, 101, 103
Renault Captur	30, 33, 36, 38, 40, 41, 44, 46, 47, 48, 50, 52, 56, 58, 60, 63, 66, 68, 70, 72, 76, 78, 80, 83, 86, 88, 90, 92, 94, 97, 98, 100, 102
Renault Scenic	30, 35, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 57, 60, 62, 64, 66, 68, 70, 71, 72, 74, 75, 77, 80, 82, 84, 86, 87, 90, 91, 94, 95, 98, 101
VW Passat	30, 35, 39, 40, 42, 45, 47, 49, 50, 52, 54, 55, 57, 60, 61, 64, 65, 67, 70, 71, 72, 73, 75, 78, 80, 81, 82, 85, 88, 90, 91, 94, 96, 98, 100

Brzine u skupu podataka se kreću u rasponu od 30 do 105 km/h, pri čemu su tačne vrijednosti prikazane u posljednjoj koloni tabele 4.1. Za niže brzine, ispod 30 km/h, kon-

trola održavanje brzine ne može funkcionisati sa odabranim vozilima (za Peugeot 3008, čak ispod 40 km/h), dok za brzine iznad 105 km/h nismo mogli izvršiti stabilna i sigurna mjerenja na odabranom putu. Korak brzine varira između 1 i 3 km/h, a sve brzine od 30 do 105 km/h su uključene u VS13 skup podataka.

4.1.2 Anotacija podataka

Originalni snimci su podeljeni na video fajlove dužine 10 sekundi (MP4 format, puna HD rezolucija, 30 frejmova u sekundi), tako da trenuci prolaska vozila budu otprilike u sredini svakog fajla. VS13 sadrži 13 foldera, sa ukupno 400 video fajlova. Svaki folder sadrži video fajlove dužine 10 sekundi i tekstualne oznake koje odgovaraju jednom vozilu. Takođe, radi praktičnosti, izdvojenis u audio snimci (frekvencija uzorkovanja: 44100 Hz, WAV format, 32-bit float PCM) i omogućili njihovo zasebno preuzimanje. Video i audio fajlove u VS13 skupu podataka prate tekstualni fajlovi sa oznakama koji sadrže dva podatka: brzinu vozila i trenutak prolaska pored kamere (sa dvocifrenom tačnošću). Trenutak prolaska pored kamere odgovara vremenskoj oznaci video kadra u kojem se vozilo počinje udaljavati iz vidnog polja kamere. Taj trenutak otprilike odgovara tački najbližeg pristupa, kako je prikazano na slici 4.1 (dolje).

Prilikom analize zvuka koji generiše vozilo tokom prolaska pored akustičnog senzora, ključno je odabrati vremenski interval u kojem su brzine najstabilnije. U ovom slučaju, interval od 3 sekunde prije trenutka prolaska i 3 sekunde poslije trenutka prolaska pored odgovarajućeg senzora je uzet u obzir prilikom selekcije uzoraka signala [12]. Ova praksa omogućava da se fokusiramo na relevantne informacije o brzini vozila i smanjimo uticaj šuma ili nepotrebnih informacija izvan tog intervala. Prikupljeni audio segmenti predstavljaju ulazne podatke za treniranje i testiranje predloženih 1D CNN modela, koji se korise za procjenu brzine vozila na osnovu akustičnih karakteristika zvuka.

4.2 Prikaz rezultata

4.2.1 Procjena brzine kao regresioni problem

U kontekstu procjene brzine vozila, RMSE se koristi kao metrika za mjerenje preciznosti modela u odnosu na stvarne brzine vozila. RMSE se računa kao kvadratni korijen

od prosječne vrijednosti kvadrata razlika između stvarnih brzina i predviđenih brzina, odnosno:

$$\text{RMSE}(y) = \sqrt{\frac{1}{N} \sum_i (f(x_i) - y_i)^2}, \quad (4.1)$$

gdje N predstavlja broj mjerenja brzine, y_i je stvarna brzina vozila u i -tom mjerenju, a $f(x_i)$ je odgovarajuća predikcija. Manje vrijednosti RMSE ukazuju na manju prosječnu grešku u predviđanju brzine vozila. To znači da model ima veću preciznost u procjeni brzine, odnosno da su predviđene vrijednosti bliske stvarnim vrijednostima brzine vozila. Kada je RMSE niska, to ukazuje na dobru sposobnost modela da se prilagodi podacima i precizno predvidi brzinu vozila na osnovu zvučnih zapisa što je važno u praktičnim scenarijima, gdje tačna procjena brzine vozila može biti ključna za različite aplikacije, kao što su sigurnost u saobraćaju ili optimizacija transportnih procesa.

Performanse procjene brzine razmatranih pristupa obuke modela, koji su prikazani u poglavlju 3.2, sa aspekta RMSE vrijednosti prikazane su u Tabeli 4.2, po vozilima i prosečno (na dnu tabele).

Tabela 4.2: RMSE vrijednosti procjene brzine

Naziv vozila	RMSE ₁ [km/h]	RMSE ₂ [km/h]
Citroen C4 Picasso	7.83	7.15
Kia Sportage	11.38	7.93
Mazda 3	9.22	8.48
Mercedes AMG550	10.44	8.5
Mercedes GLA	11.32	8.24
Nissan Qashqai	10.17	5.53
Opel Insignia	8.00	9.75
Peugeot 208	5.81	4.71
Peugeot 3008	10.70	10.25
Peugeot 307	9.49	8.09
Renault Captur	7.94	8.23
Renault Scenic	15.38	19.96
VW Passat	7.85	8.62
Srednja vrijednost	9.65	8.88

U tabeli 4.2 su prikazane vrijednosti RMSE za svako vozilo, pri čemu svaki red predstavlja jedno vozilo, a kolone su označene kao $RMSE_1$ i $RMSE_2$. Ove vrijednosti predstavljaju prosječnu grešku u predviđanju brzine vozila koristeći pristup baziran na određivanju optimalne epohe (vidjeti poglavlje 3.2.1) i pristup baziran na minimalnom gubitku na validacionom skupu (vidjeti poglavlje 3.2.2). Takođe, u posljednjem redu tabele prikazana je prosječna greška za sva vozila. Prosječna greška procjene brzine iznosi 9.65 km/h za pristup baziran na određivanju optimalne epohe i 8.88 km/h za model sa minimalnim gubitkom na validacionom skupu. Može se uočiti da drugi pristup ima manju prosječnu grešku u predviđanju brzine vozila u odnosu na prvi pristup. Osim toga, primjetno je da vozilo *Renault Scenic* ima značajno lošije performanse u procjeni brzine u poređenju sa ostalim vozilima za oba pristupa. Ovaj rezultat je očekivan jer je snimanje sesije vozila *Renault Scenic* izvršeno po vjetrovitom vremenu [5]. S druge strane, vozilo *Peugeot 208* ima najmanju grešku procjene brzine u oba slučaja jer u snimku vozila ima minimalnog šuma. Na osnovu dobijenih rezultata, možemo zaključiti da prisustvo šuma u zvučnom zapisu vozila može značajno uticati na preciznost procjene brzine vozila.

4.2.2 Procjena brzine kao klasifikacioni problem

Klasifikacija brzine vozila je proces u kojem se vrši diskretizacija opsega brzina, od najmanje do najveće iz skupa podataka, i procjenjuje se kojem opsegu pripada tačna brzina. Naime, cjelokupan opseg brzina je podijeljen u 8 klasa, tako da se brzine od 25 do manje od 35 km/h svrstavaju u prvu klasu, brzine od 35 do manje od 45 km/h se svrstavaju u drugu klasu, brzine od 45 do manje od 55 km/h u treću klasu, i tako dalje. Dakle, svaki interval predstavlja raspon brzina koji će biti klasifikovan u određenu klasu. Tačnost klasifikacije brzine je prikazana kao vjerovatnoća pravilne procjene klase.

Postupak određivanja tačnosti klasifikacije se bazira na određivanju klase brzine na osnovu zadate brzine i intervala brzina. Najprije se predviđene brzine grupišu u definisane intervale i određuju se predviđene klase. Zatim se vrši upoređivanje predviđenih klasa sa stvarnim klasama kako bi se izračunala tačnost klasifikacije. U stvarnim okolnostima, nije realno očekivati apsolutnu podudarnost između predviđene klase i stvarne klase, te se uvodi parametar Δ kako bi se uzela u obzir moguća odstupanja u klasifikaciji. Uvođenjem parametra Δ omogućava se prilagođavanje tolerancije klasifikacije na ta odstupanja. Parametar Δ definiše granicu ili prag unutar kojeg se predviđena klasa smatra ispravnom. Ako je razlika između predviđene i stvarne klase unutar Δ granice, predviđanje se smatra

tačnim, dok se u slučaju razlike između klasa veće od Δ predviđanje se smatra netačnim.

Tabela 4.3 prikazuje vjerovatnoću tačnog predviđanja klase vozila koja je Δ klasa udaljena od prave klase, koristeći metod sa optimalnom epohom. Svaka kolona predstavlja različitu vrijednost parametra Δ , koja određuje toleranciju za odstupanje između predviđene klase i stvarne klase. U drugoj koloni Tabele 4.3 prikazana je tačnost predviđanja klase kada je $\Delta = 0$, odnosno kada je predviđena klasa ista kao stvarna klasa. U koloni sa $\Delta = 1$ prikazana je vjerovatnoća tačnog predviđanja klase koja je tačno jednu klasu udaljena od stvarne klase. Slično tome, u koloni $\Delta = 2$ prikazana je vjerovatnoća tačnog predviđanja klase koja je tačno dvije klase udaljena od stvarne klase. Posljednja kolona Tabele 4.3 prikazuje vjerovatnoću tačnog predviđanja klase koja se nalazi najviše jednu klasu udaljenu od stvarne klase.

Tabela 4.3: Vjerovatnoća predviđanja klase koja je Δ klasa udaljena od prave klase - metod sa optimalnom epohom

Naziv vozila	$\Delta = 0$	$\Delta = 1$	$\Delta = 2$	$\Delta > 2$	$\Delta \leq 1$
Citroen	43.48%	43.48%	13.04%	0.00%	86.96%
Kia Sportage	24.24%	66.67%	9.09%	0.00%	90.91%
Mazda 3	62.5%	34.37%	3.13%	0.00%	96.88%
Mercedes AMG550	43.33%	43.33%	13.33%	0.00%	86.67%
Mercedes GLA	50.00%	41.18%	8.82%	0.00%	91.18%
Nissan Qashqai	34.48%	62.07%	3.44%	0.00%	96.56%
Opel Insignia	29.63%	55.56%	14.81%	0.00%	85.19%
Peugeot 208	51.72%	44.82%	3.45%	0.00%	96.55%
Peugeot 3008	25.81%	45.16%	22.58%	6.45%	70.97%
Peugeot 307	34.48%	62.07%	0.00%	3.44%	96.55%
Renault Captur	30.30%	60.61%	9.09%	0.00%	90.91%
Renault Scenic	37.14%	51.43%	11.43%	0.00%	88.57%
VW Passat	40.00%	48.57%	11.43%	0.00%	88.57%
Srednja vrijednost	39.01%	50.72%	9.51%	0.76%	89.73%

Srednje vrijednosti svake kolone su prikazane u posljednjem redu Tabele 4.3. Dakle, srednja vrijednost vjerovatnoće tačnog predviđanja klase ($\Delta = 0$) za sve automobile iznosi 39.01%, srednja vrijednost vjerovatnoće predviđanja klase koja je jednu klasu udaljena ($\Delta = 1$) je 50.72%, srednja vrijednost vjerovatnoće predviđanja klase koja je dvije klase

udaljena ($\Delta = 2$) iznosi 9.51%, srednja vrijednost vjerovatnoće predviđanja klase koja je više od dvije klase udaljena ($\Delta > 2$) je 0.76%, dok srednja vrijednost vjerovatnoće predviđanja klase koja je jednaka ili manje od jedne klase udaljena ($\Delta \leq 1$) iznosi 89.73%. Vozilo marke *Peugeot 3008* postiže najmanju vrijednost vjerovatnoće tačnog predviđanja klase koja je jednaka ili manje od jedne klase udaljena i ona iznosi 70.97%, dok ostala vozila imaju tačnost predviđanja veću od 85%.

Rezultati klasifikacije korišćenjem pristupa sa minimalnim gubitkom na validacionom skupu prikazani su u Tabeli 4.4. U poređenju sa metodom sa optimalnom epochom, ovaj pristup postiže bolje rezultate. Srednja vrijednost vjerovatnoće tačnog predviđanja klase koja je jednaka ili manje od jedne klase udaljena ($\Delta \leq 1$) iznosi 92.47%.

Tabela 4.4: Vjerovatnoća predviđanja klase koja je Δ klasa udaljena od prave klase - metod sa najmanjom greškom validacije

Naziv vozila	$\Delta = 0$	$\Delta = 1$	$\Delta = 2$	$\Delta > 2$	$\Delta \leq 1$
Citroen	43.48%	56.52%	0.00%	0.00%	100.00%
Kia Sportage	27.27%	66.67%	6.06%	1.42%	93.94%
Mazda3	53.13%	43.75%	3.13%	0.00%	96.88%
Mercedes AMG550	43.33%	56.67%	0.00%	0.00%	100.00%
Mercedes GLA	47.06%	47.06%	5.88%	1.42%	94.12%
Nissan Qashqai	75.86%	24.14%	0.00%	0.00%	100.00%
Opel Insignia	29.63%	55.56%	14.81%	0.00%	85.19%
Peugeot 208	62.07%	31.03%	6.90%	0.00%	93.10%
Peugeot 3008	38.71%	41.94%	16.13%	3.23%	80.65%
Peugeot 307	51.72%	41.38%	6.90%	0.00%	93.10%
Renault Captur	48.48%	42.42%	9.09%	0.00%	90.91%
Renault Scenic	40.00%	40.00%	17.14%	2.86%	80.00%
VW Passat	40.00%	54.29%	5.71%	1.42%	94.29%
Srednja vrijednost	46.21%	46.26%	7.06%	0.80%	92.47%

Dakle, model je u većini slučajeva uspješno predvidio klasu vozila koja je tačno jednaka ili najviše jednu klasu udaljena od stvarne klase. Osim toga, ovaj pristup je postigao tačnost predviđanja klase koja je jednaka ili manje od jedne klase udaljena ($\Delta \leq 1$) od 100% za dva vozila iz skupa podataka, a to su *Citroen* i *Mercedes AMG550*. Ostala vozila imaju tačnost predviđanja jednaku ili veću od 80%.

Glava 5

Zaključak

Zvuk koji generiše vozilo prilikom prolaska pored akustičnog senzora može se koristiti kao alternativa tradicionalnim sistemima za procjenu brzine vozila. Ova metoda je posebno značajna zbog svoje jednostavnosti i relativno niskih troškova u poređenju sa drugim tehnikama za mjerenje brzine. Konvolucione neuralne mreže mogu se uspješno koristiti za ekstrakciju relevantnih zvučnih karakteristika i postižu zadovoljavajuću preciznost u kontekstu procjene brzine vozila na osnovu zvuka.

Predloženi 1D CNN model za procjenu brzine vozila koristi neobrađeni zvučni signal i daje prosječnu grešku procjene od 8.88 km/h na VS13 skupu podataka. U poređenju sa [12], koji daje grešku procjene od 7.39 km/h, dobijen na podskupu od 10 od 13 vozila iz VS13, pristup sa minimalnim gubitkom na validacionom skupu ima primjetno veću grešku procjene. S druge strane, prednost korišćenja konvolucionih mreža je evidentna jer ne zahtijevaju ručno kreirane karakteristike, poput MA karakteristike koja je predstavljena u [12]. U slučaju predloženog 1D CNN modela, podaci ulaze direktno u neuralnu mrežu, prolazeći kroz slojeve konvolucije, sažimanja i potpuno povezanih slojeva, sve do izlaza koji predstavlja predviđenu brzinu. Ovaj pristup omogućava modelu da nauči i samostalno izvlači relevantne karakteristike iz podataka, bez potrebe za ručnim definisanjem ili izdvajanjem tih karakteristika. Time se olakšava proces treniranja i povećava fleksibilnost modela u rješavanju složenih problema. Dobijeni rezultati su obećavajući i postavljaju osnovu za buduća istraživanja.

Rezultati dobijeni u okviru magistarskog rada pokazuju da prisustvo šuma u zvučnom zapisu može narušiti performanse CNN mreža u procjeni brzine vozila na osnovu zvuka. Šum može dovesti do gubitka informacija ili izobličenja zvučnih karakteristika vozila

koje su ključne za preciznu procjenu brzine. Stoga je neophodno dodatno istražiti tehnike koje bi mogle ublažiti ovaj problem, poput augmentacije podataka, regularizacije ili prilagođavanja arhitekture CNN mreže.

Glavno ograničenje predložene metode za procenu brzine zvuka je veličina trening skupa podataka. U kontekstu dubokog učenja, skup od 400 uzoraka je mali da bi se postigli pouzdani rezultati. Tehnike unakrsne validacije mogu pomoći do određene mjere da se prevaziđe ovo ograničenje, ali je ipak neophodno znatno povećanje skupa podataka da bi se ostvario značajan napredak. U tom smislu, jedan od smjerova budućih istraživanja će biti proširenje skupa podataka korišćenjem tehnika za augmentaciju podataka. Takođe biće razmotreno povećanje skupa podataka dodavanjem uzoraka iz drugog skupa podataka koji su korišćeni u ranijim istraživanjima o brojanju vozila [46,47]. Ti uzorci bi mogli biti anotirani korišćenjem metoda za procenu brzine zasnovanih na vizuelnim informacijama. Dalje istraživanje može obuhvatiti i uključenje frekvencijskih karakteristika zvučnog signala u proces učenja, tj. jedan vid više-domenskog učenja, u cilju poboljšanja tačnosti procjene brzine vozila. Pored toga, moguće je realizovati i kombinovano učenje na osnovu vremenske reprezentacije signala i ranije predložene MA karakteristike signala.

Literatura

- [1] G. Radivojević, G. Šormaz, and B. Lazić, “Trends in development and integration of ITS,” *Journal of Road and Traffic Engineering*, vol. 66, no. 3, pp. 33–40, Oct. 2020. [Online]. Available: <https://www.putisaobračaj.rs/index.php/PiS/article/view/151>
- [2] C. Wilson, C. Willis, J. K. Hendrikz, R. Le Brocque, and N. Bellamy, “Speed cameras for the prevention of road traffic injuries and deaths,” *Cochrane database of systematic reviews*, no. 11, 2010.
- [3] M. Al-Smadi, K. Abdulrahim, and R. Abdul Salam, “Traffic surveillance: A review of vision based vehicle detection, recognition and tracking,” *International Journal of Applied Engineering Research*, vol. 11, no. 1, pp. 713–726, 2016.
- [4] D. Fernandez Llorca, A. Hernandez Martinez, and I. Garcia Daza, “Vision-based vehicle speed estimation: A survey,” *IET Intelligent Transport Systems*, vol. 15, no. 8, pp. 987–1005, 2021. [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/itr2.12079>
- [5] S. Djukanović, N. Bulatović, and I. Čavor, “A dataset for audio-video based vehicle speed estimation,” in *2022 30th Telecommunications Forum (TELFOR)*, 2022, pp. 1–4.
- [6] M. Abbasi, A. Shahraki, and A. Taherkordi, “Deep learning for network traffic monitoring and analysis (ntma): A survey,” *Computer Communications*, vol. 170, pp. 19–41, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0140366421000426>
- [7] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, “Cnn variants for computer vision: History, architecture, application, challenges and future scope,” *Electronics*, vol. 10, no. 20, p. 2470, 2021.

- [8] S. Barnwal, R. Barnwal, R. Hegde, R. Singh, and B. Raj, “Doppler based speed estimation of vehicles using passive sensor,” in *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. IEEE, 2013, pp. 1–4.
- [9] H. V. Koops and F. Franchetti, “An ensemble technique for estimating vehicle speed and gear position from acoustic data,” in *2015 IEEE International Conference on Digital Signal Processing (DSP)*. IEEE, 2015, pp. 422–426.
- [10] E. Kubera, A. Wiczorkowska, A. Kuranc, and T. Słowik, “Discovering speed changes of vehicles from audio data,” *Sensors*, vol. 19, no. 14, p. 3067, 2019.
- [11] J. Giraldo-Guzmán, A. G. Marrugo, and S. H. Contreras-Ortiz, “Vehicle speed estimation using audio features and neural networks,” in *2016 IEEE ANDESCON*. IEEE, 2016, pp. 1–4.
- [12] S. Djukanović, J. Matas, and T. Virtanen, “Acoustic vehicle speed estimation from single sensor measurements,” *IEEE Sensors Journal*, vol. 21, no. 20, pp. 23 317–23 324, 2021.
- [13] N. Bulatović and S. Djukanović, “An approach to improving sound-based vehicle speed estimation,” in *2022 Zooming innovation in consumer technologies conference (ZINC 2022)*. IEEE, 2022.
- [14] N. Bulatović and S. Djukanović, “Mel-spectrogram features for acoustic vehicle detection and speed estimation,” in *2022 26th International Conference on Information Technology (IT)*. IEEE, 2022, pp. 1–4.
- [15] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1d convolutional neural networks and applications: A survey,” *Mechanical systems and signal processing*, vol. 151, p. 107398, 2021.
- [16] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
- [17] A. Ng. (2017–2018) Deep learning specialization. [Online]. Available: <https://www.coursera.org/specializations/deep-learning>
- [18] A. Ng, “Neural networks and deep learning,” <https://www.coursera.org/learn/neural-networks-deep-learning>, 2018.

- [19] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, E. Muhamagic, R. Parekh, and C. becoming an effective contributor to an open-source software project, “Deep learning applications and challenges in big data analytics,” *Journal of Big Data*, vol. 2, no. 1, pp. 1–21, 2015.
- [20] I. J. Goodfellow, Q. V. Le, A. M. Saxe, H. Lee, and A. Y. Ng, “Measuring invariances in deep networks,” *arXiv preprint arXiv:1312.6034*, 2013.
- [21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [22] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 448–456. [Online]. Available: <https://proceedings.mlr.press/v37/ioffe15.html>
- [23] A. Karpathy, J. Johnson, and L. Fei-Fei. (2019) Cs231n: Convolutional neural networks for visual recognition. Stanford University. Accessed: 2023-05-09. [Online]. Available: <http://cs231n.stanford.edu/>
- [24] S. M. Warren and P. Walter, “A logical calculus of the ideas immanent in nervous activity,” *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [25] K. Fukushima and S. Miyake, “Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position,” *Pattern Recognition*, vol. 15, no. 6, pp. 455–469, 1982. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0031320382900243>
- [26] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, “Handwritten digit recognition with a back-propagation network,” in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 2. Morgan-Kaufmann, 1989. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf
- [27] L. Deng, “The mnist database of handwritten digit images for machine learning research,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

- [28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- [29] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv 1409.1556*, 09 2014.
- [30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [32] X. Chen, F. Kopsaftopoulos, Q. Wu, H. Ren, and F.-K. Chang, “A self-adaptive 1d convolutional neural network for flight-state identification,” *Sensors*, vol. 19, no. 2, p. 275, Jan 2019.
- [33] F. Paissan, V. P. Kumaravel, and E. Farella, “Interpretable cnn for single-channel artifacts detection in raw eeg signals,” in *2022 IEEE Sensors Applications Symposium (SAS)*, Aug 2022.
- [34] U. R. Acharya and et al., “A deep convolutional neural network model to classify heartbeats,” *Computers in Biology and Medicine*, vol. 89, pp. 389–396, Oct 2017.
- [35] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, and D. J. Inman, “Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks,” *Journal of Sound and Vibration*, vol. 388, pp. 154–170, Feb 2017.
- [36] X. Yuan, D. Tanksley, L. Li, H. Zhang, G. Chen, and D. Wunsch, “Faster post-earthquake damage assessment based on 1d convolutional neural networks,” *Applied Sciences*, vol. 11, no. 21, p. 9844, Oct 2021.
- [37] L. Eren, “Bearing fault detection by one-dimensional convolutional neural networks,” *Mathematical Problems in Engineering*, vol. 2017, pp. 1–9, 2017.

- [38] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, “Real-time motor fault detection by 1-d convolutional neural networks,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, Nov 2016.
- [39] C. Du and et al., “Research on fault diagnosis of automobile engines based on the deep learning 1d-cnn method,” *Engineering Research Express*, vol. 4, no. 1, p. 015003, Jan 2022.
- [40] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, “A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals,” *Sensors*, vol. 17, no. 2, 2017. [Online]. Available: <https://www.mdpi.com/1424-8220/17/2/425>
- [41] L. Eren, T. Ince, and S. Kiranyaz, “A generic intelligent bearing fault diagnosis system using compact adaptive 1d cnn classifier,” *J. Signal Process. Syst.*, vol. 91, pp. 179–189, 2019.
- [42] LJ. Stanković and D. Mandić, “Convolutional neural networks demystified: A matched filtering perspective-based tutorial,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.
- [43] B. Alsallakh, N. Kokhlikyan, V. Miglani, J. Yuan, and O. Reblitz-Richardson, “Mind the pad—cnn’s can develop blind spots,” *arXiv preprint arXiv:2010.02178*, 2020.
- [44] C. Mouton, J. C. Myburgh, and M. H. Davel, “Stride and translation invariance in cnn’s,” pp. 267–281, 2020.
- [45] J. Brownlee, “How to use learning curves to diagnose machine learning model performance,” *Deep Learning Performance*, February 2019, last Updated on August 6, 2019.
- [46] S. Djukanović, J. Matas, and T. Virtanen, “Robust audio-based vehicle counting in low-to-moderate traffic flow,” in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1608–1614.
- [47] S. Djukanović, Y. Patel, J. Matas, and T. Virtanen, “Neural network-based acoustic vehicle counting,” in *29th European Signal Processing Conference (EUSIPCO 2021)*, 2021.