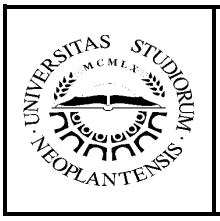


УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА

УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
НОВИ САД
Департман за рачунарство и аутоматику
Одсек за рачунарску технику и рачунарске комуникације

МАСТЕР РАД

Кандидат: **Алекса Арсић**
Број индекса: **E2 134-2020**

Тема рада: **Једно решење адаптивног темпомата коришћењем машинског учења унутар КАРЛА симулатора**

Ментор рада: **доц. др Богдан Павковић**

Нови Сад, септембар, 2022.

УНИВЕРЗИТЕТ У НОВОМ САДУ ● **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, **РБР**:		
Идентификациони број, **ИБР**:		
Тип документације, **ТД**:		Монографска документација
Тип записа, **ТЗ**:		Текстуални штампани материјал
Врста рада, **ВР**:		Мастер рад
Аутор, **АУ**:		**Алекса Арсић**
Ментор, **МН**:		**доц. др Богдан Павковић**
Наслов рада, **НР**:		**Једно решење адаптивног темпомата коришћењем машинског учења унутар КАРЛА симулатора**
Језик публикације, **ЈП**:		Српски / ћирилица
Језик извода, **ЈИ**:		Српски
Земља публиковања, **ЗП**:		Република Србија
Уже географско подручје, **УГП**:		Војводина
Година, **ГО**:		**2022.**
Издавач, **ИЗ**:		Ауторски репринт
Место и адреса, **МА**:		Нови Сад; трг Доситеја Обрадовића 6
Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога)		**8/64/21/1/25/2/0**
Научна област, **НО**:		Електротехника и рачунарство
Научна дисциплина, **НД**:		Рачунарска техника
Предметна одредница/Кључне речи, **ПО**:		**Машинско учење, дубоко учење, вештачке неуронске мреже, конволутивне неуронске мреже, обрада слике, систему аутоматског управљања, одржавање растојања**
УДК		
Чува се, **ЧУ**:		У библиотеци Факултета техничких наука, Нови Сад
Важна напомена, **ВН**:		
Извод, **ИЗ**:		**Систем за адаптивно одржавање растојања заснован на обради слике уз помоћ принципа машинског учења и система аутоматског управљања унутар КАРЛА симулатора за симулације саобраћајних ситуација. У раду је приказана теоријска позадина овог система, као и програмско решење и постигнути резултати.**
Датум прихватања теме, **ДП**:		
Датум одбране, **ДО**:		
Чланови комисије, **КО**:	Председник:	доц. др Наташа Самарџић
	Члан:	доц. др Жељко Лукач
	Члан, ментор:	доц. др Богдан Павковић

Потпис ментора

Accession number, **ANO**:			
Identification number, **INO**:			
Document type, **DT**:		Monographic publication	
Type of record, **TR**:		Textual printed material	
Contents code, **CC**:		Master Thesis	
Author, **AU**:		**Aleksa Arsić**	
Mentor, **MN**:		**Bogdan Pavković, PhD**	
Title, **TI**:		**One solution of adaptive cruise control system using machine learning inside CARLA simulator**	
Language of text, **LT**:		Serbian / cyrillic	
Language of abstract, **LA**:		Serbian	
Country of publication, **CP**:		Republic of Serbia	
Locality of publication, **LP**:		Vojvodina	
Publication year, **PY**:		2022.	
Publisher, **PB**:		Author's reprint	
Publication place, **PP**:		Novi Sad, Dositeja Obradovica sq. 6	
Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes)		**8/64/21/1/25/2/0**	
Scientific field, **SF**:		Electrical Engineering	
Scientific discipline, **SD**:		Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, **S/KW**:		Machine learning, deep learning, artificial neural networks, convolutional neural networks, image processing, control systems, adaptive cruise control	
UC			
Holding data, **HD**:		The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, **N**:			
Abstract, **AB**:		**Adaptive cruise control system based on image processing with machine learning principals, convolutional neural networks and control systems inside CARLA simulator. In this paper, theoretical background of the system is shown, adaptive cruise control implementation, testing methods and results.**	
Accepted by the Scientific Board on, **ASB**:			
Defended on, **DE**:			
Defended Board, **DB**:	President:	Nataša Samardžić, PhD	
	Member:	Željko Lukač, PhD	Menthor's sign
	Member, Mentor:	Bogdan Pavković, PhD	

Захвалност

Велику захвалност дугујем ментору, др Богдану Павковићу који је својом стручношћу и стрпљењем потпомогао изради овог рада, те изузетном стручњаку у пољу машинског учења и великом пријатељу овога рада, др Велибору Илићу.

Највише се захваљујем мојим најмилијима – мајци, породици, пријатељима и свим драгим људима који су ми пружили несебичну подршку и разумевање током како основних, тако и мастер студија. Хвала вам!

Садржај

Захвалност 3
Садржај 4
Списак слика 6
Списак табела 8
Списак графикона 9
Списак формула 10
1 Увод 10
2 Преглед литературе 13
3 Теоријске основе 15
3.1 Машинско учење 15
3.2 Дубоко учење 16
3.2.1 Вештачке неуронске мреже 17
3.2.2 Примене дубоког учења 18
3.3 Конволутивне неуронске мреже 19
3.3.1 Улаз у мрежу 20
3.3.2 Конволутивни слој 21
3.3.3 Сажимајући слој 23
3.3.4 Потпуно повезани слој 24
3.3.5 Обучавање, параметри модела и хипер-параметри 25
3.4 Аутоматско управљање системима 27
3.4.1 ПИД контролер 28

3.5 КАРЛА симулатор 30
4 Теоријске основе програмског решења 33
4.1 Конволутивна неуронска мрежа 34
4.1.1 Интерпретација излазних параметара 35
4.1.2 Процес обучавања неуронске мреже 36
4.1.2.1 Метрика тачности валидације 37
4.2 Адаптивни блок 38
4.3 Лонгитудинални ПИД контролер 39
4.3.1 Наштимавање коефицијената K_p, K_i и K_d 39
4.4 Систем лонгитудиналне контроле аутомобила 41
5 Програмско решење 43
5.1 КАРЛА симулациони свет 44
5.2 КАРЛА сценарио 46
6 Резултати 50
6.1 Резултати обучавања конволутивне неуронске мреже 50
6.2 Резултати система за адаптивно одржавање растојања 52
6.2.1 Први сим. случај – празна цеста 52
6.2.2 Други сим. случај – текући саобраћај (са водећим возилом) 53
6.2.3 Трећи сим. случај – текући саобраћај (без водећег возила) 54
6.2.4 Четврти сим. случај – возило се престројава у траку его возила 54
6.2.5 Пети сим. случај – возило се нагло престројава у траку его возила 56
6.2.6 Шести сим. случај – его возило наилази на статички саобраћај 57
7 Закључак 58
8 Референце 60

Списак слика

Слика 3.1 Структура дубоке неуронске мреже [10] 17
Слика 3.2 Пример архитектуре конволутивне неуронске мреже [14] 19
Слика 3.3 Преглед РГБ канала 20
Слика 3.4 Приказ операције конволуције [15] 21
Слика 3.5 Пример *Max-pool* и *Average-pool* метода са кораком = 2 23
Слика 3.6 Потпуно повезани слој вештачких неуронских мрежа 24
Слика 3.7 Нежељена и жељена стања модела неуронских мрежа [16] 26
Слика 3.8 Основна контролна петља система аутоматског управљања 27
Слика 3.9 Контролна петља са ПИД контролером 29
Слика 3.10 КАРЛА сервер 31
Слика 3.11 КАРЛА клијент 31
Слика 4.1 Контролна петља система за адаптивно одржавање растојања 33
Слика 4.2 Архитектура конволутивне неуронске мреже 34
Слика 4.3 Интерпретација распона вредности параметра D_{acc} 36
Слика 4.4 Процес обучавања неуронске мреже 37
Слика 5.1 Ток података унутар КАРЛА симулационог света 43
Слика 5.2 Ток података унутар КАРЛА симулационог света 2 44
Слика 5.3 Ток података унутар КАРЛА сценарија 46
Слика 5.4 Регија од интереса конволутивне неуронске мреже 47
Слика 6.1 Симулациони случај – празна цеста 52
Слика 6.2 Симулациони случај – текући саобраћај, са водећим возилом 53

Слика 6.3 Симулациони случај – текући саобраћај, без водећег возила 54
Слика 6.4 Симулациони случај – возило се престројава у траку его возила.....55
Слика 6.5 Симулациони случај – возило се нагло престројава у траку его возила56
Слика 6.6 Симулациони случај – статички саобраћај.....57

Списак табела

Табела 4.1 Наштимани коефицијенти ПИД регулатора 41

Списак графикона

Графикон 6.1 Рез. тачности обучавања и валидације модела конв. неур. мр.....51
Графикон 6.2 Губици приликом обучавања конволутивне неуронске мреже 51

Списак формула

(3.1) Формула операције конволуције неуронске мреже 21
(3.2) Формула РЕЛУ активационе функције 22
(3.3) Формула излазне димензије конволутивног слоја неуронске мреже 22
(3.4) Формула броја елемената кернела неуронске мреже 22
(3.5) Формула броја тежниских коефицијената неуронске мреже 24
(3.6) Формула преносне карактеристике основне контролне петље САУ 28
(3.7) Формула ПИД контролера 1 30
(4.1) Излазни тензор конволутивне неуронске мреже 35
(4.2) Формула метрике тачности валидације 38
(4.3) Формула ПИД контролера 2 39
(4.4) Мрежа наштимавања коефицијената ПИД контролера 40
(4.5) Формула функције доброте 40
(4.6) Формула времена извршавања наштимавања коеф. ПИД контр. 41
(5.1) Формула *Simple Moving Average* 47
(5.2) Формула стопе промене промењиве 48

1 Увод

Експоненцијална експанзија технологије и електронике у двадесет и првом веку допринела је великом ослањању просечног човека на њене способности. Од мобилних телефона, медицинских уређаја, преко потрошачке електронике и мултимедије, па тако и аутомобилске индустрије, људи се све више ослањају на наменске уграђене системе који им олакшавају свакодневни живот. Рапидни развој аутомобилске индустрије јавља се као одговор инжењерске заједнице на питање безбедности саобраћаја, самог возила и његових путника, али и условно побољшање људских могућности и способности приликом управљања возилима. Па тако, развијају се напредни системи за асистенцију возачу (енг. *Advanced Driver Assistance System – ADAS*) који служе управо томе.

У аутомобилима, све је већи број система који олакшавају њихово управљање, где за пример можемо узети адаптивни темпомат, системе за асистирање приликом паркирања, али и системе који под одређеним околностима самостално управљају возилом. Пример система који самостално управља возилом по отвореном путу, али очекује да човек интервенише у случају да сам систем није способан донети одлуку, јесте ауто пилот систем компаније Тесла, уграђен у истоимене аутомобиле. Ауто пилот компаније Тесла ослања се на принципе машинског учења, где у свом решењу користе вештачку неуронску мрежу која као улазе прима фотографије са осам камера истовремено.

Машинско учење (енг. *Machine learning*) представља групу специфичних алгоритама који рачунарима омогућавају да на основу стеченог искуства доносе одлуке без да су експлицитно програмирани за то. У алгоритме машинског учења спадају и

вештачке неуронске мреже коју сачињавају скупови вештачких неурона који су повезани конекцијама које прослеђују сигнале до других вештачких неурона, а те конекције још се називају и синапсе. Вештачки неурон који прими сигнал га обрађује и прослеђује другим вештачким неуронима са којима је повезан, а излазни сигнал који се прослеђује израчунава се уз помоћ нелинеарне математичке функције. Обучавањем вештачких неуронских мрежа формира се математички модел који се води искуством и ако је добро формиран може да предвиди шта би требало да се појави на излазу неуронске мреже без да је такав случај видео приликом обучавања. Само обучавање врши се уз помоћ сета података (енг. *Dataset*) који неуронској мрежи омогућава да формира потребан математички модел. Архитектура вештачких неуронских мрежа постоји мноштво, међутим за обраду фотографије користе се конволутивне неуронске мреже које чине основу овог рада.

Поред механизма обраде фотографије и екстракције потребних информација, за деловање на механичке системе, па тако и системе управљања аутомобила, потребно је искористити и методе система аутоматског управљања. Систем аутоматског управљања, уз помоћ контролера има могућност сталног поређења жељених излазних резултата са тренутним резултатима, где се примењује сигнал разлике како би се систем довео у жељено стање.

Овај рад описује једно решење система за адаптивно одржавање растојања коришћењем техника машинског учења и конволутивних неуронских мрежа као механизма обраде фотографије и коришћењем теорије система аутоматског управљања како би се деловало на механичке делове аутомобила и тиме се у потпуности контролисао без интервенције човека. Систем се састоји од конволутивне неуронске мреже која има способност препознавања да ли се на улазној слици налазе возила, диференцирања у којој се возној траци налазе возила и способност давања степена поузданости одлуке да ли је потребно убрзати или успорити наше возило или је потребно одржавати растојање између нашег возила и возила које се налази у истој траци испред нашег возила. Поред тога, систем чине и адаптивни блок за постављање жељене брзине кретања аутомобила, али и ПИД контролер који тежи успоставити жељену брзину кретања.

Поставка рада, реализована је у КАРЛА (енг. *CARLA*) симулатору отвореног кода, који омогућава креирање реалних саобраћајних ситуација, али и добављање корисних информација о учесницима у саобраћају, кроз кориснички интерфејс уз помоћ

програмског језика Питон (енг. *Python*). Поред тога, пројектовање и обучавање конволутивне неуронске мреже реализовано је уз помоћ *Tensorflow 2.0* Питон библиотеке која у себи садржи *Keras* модул потребан за дефинисање њеног модела.

У наставку је дат кратак опис структуре мастер рада:

2. **Преглед литературе** – дат је преглед литературе, мотивација за писање рада и јединствености овог решења система за адаптивно одржавање растојања у односу на различите радове на тему.
3. **Теоријске основе** – у овом поглављу дат је кратак преглед теоријских основа машинског учења, дубоког учења, конволутивних неуронских мрежа, система аутоматског управљања, ПИД контролера и КАРЛА симулатора.
4. **Теоријске основе програмског решења** – у овом поглављу описани су теоријски концепти потребни за разумевање и имплементацију овог решења система за адаптивно држање растојања, а у које спадају: контролна петља система, процес пројектовања и обучавања неуронских мрежа и теорија примене ПИД контролера у контролној петљи.
5. **Програмско решење** – у овом поглављу дат је опис програмског решења са два нивоа апстракције; са становишта КАРЛА симулационог света и са становишта КАРЛА сценарија. Описане су битне структуре и токови података кроз програмско решење и контролну петљу система. Такође, представљен је концепт *SimScenarioRunner* механизма који омогућава извршавање више узастопних КАРЛА сценарија.
6. **Резултати** – у овом поглављу разматрају се постигнути резултати система за адаптивно одржавање растојања на нивоу елемената контролне петље. Представљени су резултати обучавања конволутивне неуронске мреже, као и понашање система унутар КАРЛА симулатора.
7. **Закључак** – у овом поглављу дат је критички осврт на целокупан рад и постигнуте резултате овог решења система за адаптивно одржавање растојања.

2 Преглед литературе

Системи који управљају лонгитудиналним контролама аутомобила не представљају новину у свету аутомобилске индустрије, те комерцијална решења вуку корене из прошлог века. Међутим, развојем технике и аутомобилске индустрије, најпопуларније поље истраживања у оквиру напредних система за асистенцију возачу представљају управо системи адаптивног одржавања растојања. Ови системи се данас могу наћи у великој примени код скоро свих произвођача аутомобила и њиховим луксузним серијама, али и у све већем броју средњих серија аутомобила. Разлог због кога су они изузетно занимљиви за проучавање, представља чињеница да они у комбинацији са другим системима у оквиру напредних система за асистенцију возачу, као што су системи за латералну контролу, дају основ аутономне вожње, чиме ова тема привлачи све већи број истраживача и произвођача аутомобила.[1]

Поред тога, принципи машинског учења и алгоритми вештачких неуронских мрежа налазе све свећу примену у разним сферама, као што су: медицина, финансије, минералогији, па тако и у аутомобилској индустрији. [2][3][4][5]

Спој аутомобилске индустрије и машинског учења, односно вештачких неуронских мрежа потенцијално представља плодно тло за развој потпуно аутономних возила, која би вођења претходним искуством увелико олакшала човекову свакодневницу, оптимизовала време проведено у путу, али и поспешила безбедност саобраћаја и његових учесника. Сходно томе, видљива је велика улога интелигентних

система за лонгитудиналну контролу возила, која у својим решењима користе вештачке неуронске мреже као градивне блокове.

У раду [6], описано је једно такво решење система за адаптивно одржавање растојања уз помоћ вештачке неуронске мреже. Мери се растојање између возила у коме је постављен систем и водећег возила, те се на основу те информације одређује жељена брзина кретања. Вештачка неуронска мрежа задужена је да, на основу информација о жељеној брзини кретања, дефинише степен притискања папучице гаса или кочнице. Аутори су дошли до закључка, на основу девијације стварне брзине од жељене брзине кретања, да су вештачке неуронске мреже способне за доношење описаних одлука са девијацијама брзине у распону од 6.8 до 11.8 километара на час, у зависности од коришћене архитектуре неуронске мреже.

Са друге стране, у раду [7] разматра се само контрола папучице кочнице. Вештачка неуронска мрежа у зависности од различитих параметара, као што су: стање површине пута, временски услови, нагиб пута, итд., даје процену заустављног пута. Процена заустављног пута користи се даље у наредној вештачкој неуронској мрежи која на основу те информације и информација о типу возила, стању кочница и оптерећења возила процењује степен потребног притискања папучице кочнице. Као позитиван закључак о коришћењу вештачких неуронских мрежа у оваквим системима аутори наводе постигнуте резултате, где су неуронске мреже постигле високе тачности над ограниченим скупом података за обучавање.

Овај рад, као што је поменуто, описује једно решење система за адаптивно одржавање растојања, уз помоћ конволутивне неуронске мреже и метода система аутоматског управљања. Поред тога, пројектовани систем има могућност покретања аутомобила од нулте брзине кретања, као и потпуно заустављање уколико је то потребно без интервенисања од стране човека. Осим тога, иако је конволутивна неуронска мрежа обучавана уз разматрање растојања између возила у коме је овај систем имплементиран и водећег возила, као и тренутне брзине кретања, приликом свог рада, као релевантан податак узима слику возног окружења, те даје повратне информације о њему без контекста брзине кретања и растојања између возила. Овим се покушало постићи и условно поспешити понашање слично човековом, који својим осетилима вида интуитивно процењује да ли је потребно кочити, убрзати или одржавати растојање.

3 Теоријске основе

Ово поглавље покрива кратке теоријске основе машинског учења, дубоког учења, конволутивних неуронских мрежа, система аутоматског управљања и КАРЛА симулатора које представљају основу овог рада.

3.1 Машинско учење

Машинско учење представља грану проучавања специфичне фамилије алгоритама која поседује могућност аутоматског побољшања излазних резултата на основу стеченог искуства. Често се за машинско учење каже да оно представља подскуп вештачке интелигенције из разлога јер алгоритми машинског учења проналазе узорке над великом количином улазних података, креирају математички модел и на основу таквог стеченог искуства могу даураде предикцију или направе одлуку без да су експлицитно програмирани за то.

Другим речима, машинско учење јесте поступак у коме рачунари откривају како да на основу великог броја података, које им дамо на располагање, генерализују некакве специфичности проблема које треба решити. Крајњи циљ јесте да се рачунарима омогући и поспеши процес аутоматског учења, али и доношење самосталних одлука без интервенције од стране човека. Генерализација, у овом смислу, јесте сама способност рачунара да даје тачне резултате над скупом података који није видео.

Подаци које шаљемо на улаз алгоритама машинског учења приликом обучавања још се називају и подаци за обучавање (енг. *Training data*). Они представљају основу за стицање искуства, тј. основу за генерализацију од стране алгоритама машинског учења.

Спектар примене ових алгоритама је веома широк и велика је могућност да их свакодневно користимо и долазимо у дотицај са њима, а да тога нисмо свесни. Алгоритми машинског учења могу се наћи као саставни део виртуалних личних асистената, могу се пронаћи као део предикција које користе системи за навигацију, у системима видео надзора, друштвеним мрежама (људи које можда познајете, препознавање лица, ...), у филтрирању електронске поште, у онлајн корисничким подршкама, онлајн детекција превара, итд.

Приступи учења задатог проблема могу се поделити у три категорије: надгледано учење, ненадгледано учење и полунадгледано учење. Надгледано учење представља облик стицања искуства од стране алгоритма машинског учења у коме се креира математички модел проблема на основу улазних података и жељених излазних података. Ненадгледано учење се базира на учењу уз помоћ скупа за обучавање који се састоји само од улаза, без назначених очекиваних излазних вредности. Полунадгледано учење представља комбинацију надгледаног учења и ненадгледаног учења. Комбинацијом ова два имамо податке за обучавање који садрже мањи број означених података, односно податке које садрже очекиване излазне вредности и податке који садрже само улазне вредности без података о излазним вредностима.

3.2 Дубоко учење

Дубоко учење произлази из шире фамилије метода машинског учења у којима се, као носиоци процеса машинског учења, примењују вештачке неуронске мреже. Вештачке неуронске мреже састављене су од више слојева вештачких неурона, који су на одговарајући начин повезани. Сваки слој састоји се од различитог броја вештачких неурона од којих сваки врши некакву линеарну или нелинеарну математичку манипулацију сигнала који долази до њега. Због такве слојевите структуре вештачких неуронских мрежа долази придев „дубоко“.

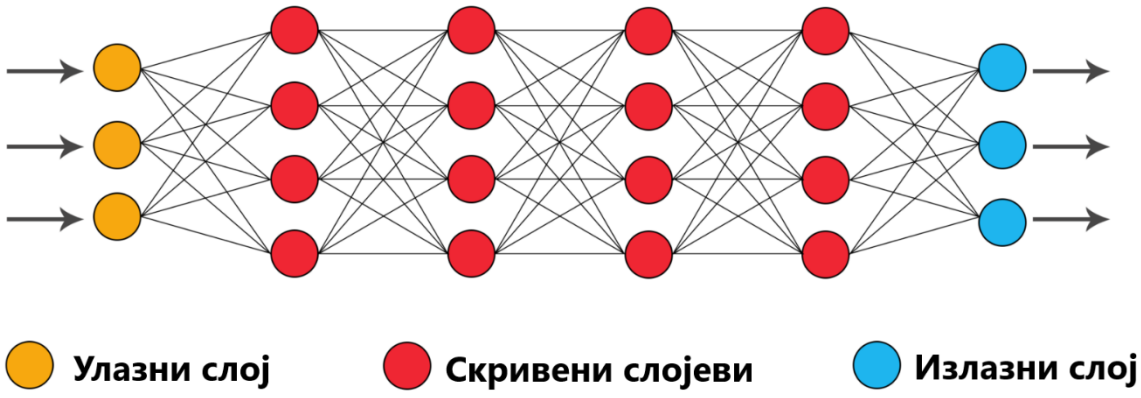
Као и код машинског учења, крајњи циљ дубоког учења јесте да се омогући и поспеши рачунарима да, на основу улазних података, генерализују специфичности некаквог проблема. Такође и неки приступи таквом учењу су им заједнички: надгледано учење, ненадгледано учење и полунадгледано учење. Код дубоког учења постоји и

учење условљавањем (енг. *Deep Reinforcement Learning*), као још један од приступа обучавања вештачких неуронских мрежа. [8]

3.2.1 Вештачке неуронске мреже

Структура дубоких неуронских мрежа је таква да сигнал пролази кроз више слојева на свом путу од улаза до излаза из мреже. Сигнал се између слојева креће преко веза које повезују неуроне различитих слојева, а још се називају и синапсе. Сваки неурон једног слоја може бити повезан са сваким неуроном следећег слоја и тада такве слојеве називамо потпуно повезаним слојевима (енг. *Fully connected layers*). Вештачка неуронска мрежа сматра се „дубоком“ ако постоје бар 3 слоја кроз која сигнал пролази, укључујући и улазни и излазни слој. Оне могу имати велики број слојева, нпр. *ResNet* са 1202 слоја. [9]

Вештачке неуронске мреже пројектоване су да препознавају заједничке особине сигнала из стварнога света из перспективе машине. Заједничке особине које оне препознавају су нумеричког типа и сви подаци из стварнога света, које желимо да пропустимо кроз вештачку неурноску мрежу, морају да буду представљени нумерички. Често, да би се избегле неправилности у раду вештачке неуронске мреже, нумерички подаци представљени су у нормализованом опсегу вредности [0, 1] или [-1, 1].



Слика 3.1 Структура дубоке неуронске мреже [10]

Сваки слој унутар мреже обучава се на посебним специфичностима сигнала у односу на излаз претходног нивоа. Што се улази дубље у мрежу, то су неурони способнији да препознају комплексније специфичности сигнала јер гомилају и рекомбинују специфичности претходног слоја.

Ако се ради о проблему класификације, проласком улазног сигнала кроз мрежу, на излазу се појављује вероватноћа некаквог излаза. На пример, ако пропустимо слику на којој се налази пас кроз, за то, обучену неурноску мрежу, она на излазу може дати вредност која представља вероватноћу да је на слици одређена пасмина.

Ако се ради о проблему регресије, проласком улазног сигнала кроз мрежу, на излазу се појављује нумеричка вредност која одговара нечему што би могло идуће да се деси, појави,... На пример, ако на улаз вештачке неуронске мреже доведемо информације о стамбеном простору, као што су: површина стамбеног простора, површина подрума, постоји ли двориште, клима, итд.; она на излазу може да предвиди његову цену.

Архитектуре вештачких неуронских мрежа долазе у разним облицима, због различитих потреба и врста проблема. Нама најзначајнија архитектура, која се користи приликом израде овог рада јесте архитектура конволутивних неуронских мрежа. Кратак теоријски преглед описан је у секцији 2.3

3.2.2 Примене дубоког учења

У наредном делу дат је кратак преглед неких примена дубоког учења, од којих класификација фотографија и аутоматско препознавање говора имају значајан ниво перформансе. [11], [12]

Класификација фотографија на бази дубоког учења последњих је година доведена до завидног нивоа чак и за човека и човекове способности. Резултати такмичења које евалуира алгоритме за детекцију и класификацију улазних фотографија (енг. *Large Scale Visual Recognition Challenge (ILSVRC)*) из 2017. године показују да је грешка класификације објеката алгоритама дубоког учења 2,3%, док је грешка код класификације објеката на фотографијама код човека 5%. [11] Дубоко учење нашло је своју примену и у препознавању говора и обично се врши на ТИМИТ акустично-континуалном говорном скупу података (енг. *TIMIT Acoustic-Phonetic Continuous Speech Corpus*). Једно од бољих решења над ТИМИТ скупом података предложено је унутар рада [12], са ПЕР грешком од 16.9%.

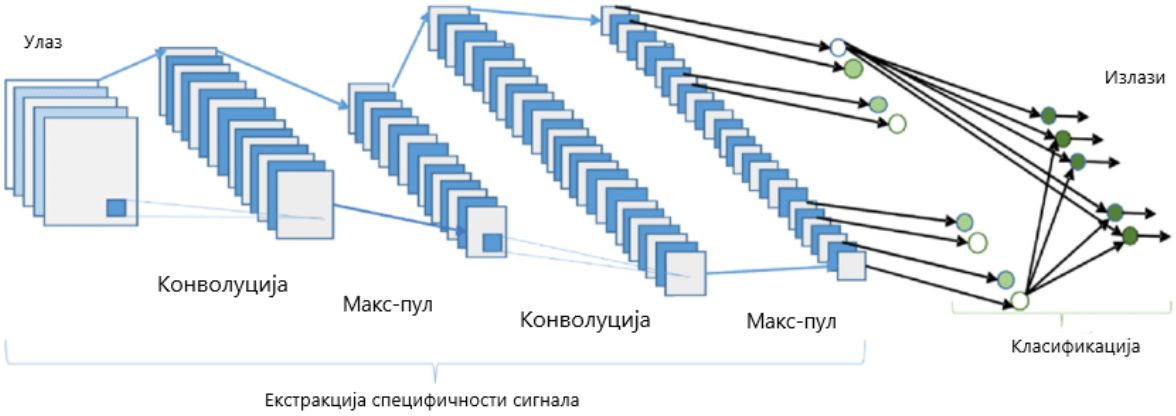
Дубоке неуронске мреже су се показале као добар алат за идентификацију стилског периода дате фотографије, као и екстракцију датог стила и његову примена над насумичном фотографијом. Такође, у могућности су да генеришу занимљиве фотографије на основу насумичних визуалних улазних параметара. Постоји и *AtomNet*

систем дубоког учења за структурно пројектовање лекова. Кориштен је за предвиђање кандидата биомолекула за болести као што је Ебола и Мултипла склероза. [13]

3.3 Конволутивне неуронске мреже

Конволутивне неуронске мреже представљају архитектуру вештачких неуронских мрежа које се најчешће примењују у пољу рачунарске визије. Једна од њихових битнијих карактеристика јесте да су транслационо инваријантне, односно имају способност да детектују исти објекат на различитим фотографијама без обзира на његов положај унутар фотографије. Да би се вештачка неуронска мрежа сматрала конволутивном мрежом, потребно је да бар један скривени слој имплементира математичку операцију конволуције.

На слици 2.5 може се видети уобичајена архитектура конволутивне неуронске мреже. Два главна дела од којих се састоји свака конволутивна неуронска мрежа су: екстракција специфичности и класификација. Ако погледамо дубље унутар дела за екстракцију специфичности можемо приметити да конволутивна неруонска мрежа имплементира два типа слојева: конволутивни (енг. *Convolution*) и макс-пул (енг. *Max-pooling*) сажимајући слој. Сви ови слојеви, који се налазе између улаза и излаза неуронске мреже су скривени слојеви. Као активацијска функција, најчешће се користи РЕЛУ (енг. *RELU, Rectified linear unit*) слој који је попраћен додатним сажимајућим слојевима, потпуно повезаним слојевима и слојевима за класификацију. Активацијски слој, односно активацијска функција омогућава неуронској мрежи да разлучи које информације су од значаја, док сви ирелевантни подаци остају неискориштени. Детаљнији описи појединачних слојева и њихових функционалности могу се наћи у наредним поглављима.

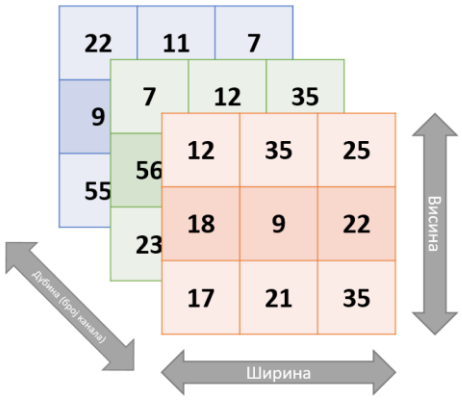


Слика 3.2 Пример архитектуре конволутивне неуронске мреже [14]

3.3.1 Улаз у мрежу

Ако се ради о слици РГБ (енг. *Red, Green, Blue - RGB*) формата улаз у конволутивну неуронску мрежу јесте слика димензија ($W \times H \times C$), а која је представљена матрицом истих димензија. Где су:

- W – висина
- H – ширина
- C – број канала (за РГБ формат – 3)

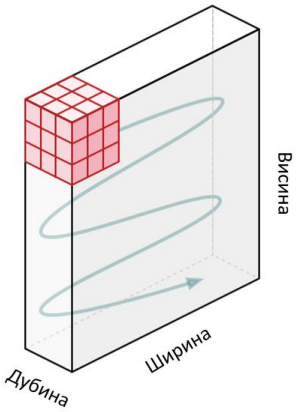


Слика 3.3 Преглед РГБ канала

Задатак конволутивне неуронске мреже јесте да слике сведе у форму која је једноставнија за обраду, али без губитака специфичности слике које су неопходне за давање добре предикције на излазу.

Формати простора боја, који се могу наћи на улазу у мрежу, су: црно-бела фотографија (енг. *Grayscale*), РГБ (енг. *Red, Green, Blue - RGB*), ХСВ (енг. *Hue, Saturation, Value - HSV*), итд.

3.3.2 Конволутивни слој



Слика 3.4 Приказ операције конволуције [15]

Са слике 2.7 се може видети операција конволуције која се врши над улазном сликом и кернелом (на слици означено црвеном бојом) чији су елементи параметри које је потребно да мрежа научи током процеса њеног обучавања. Кернел се по слици, у зависности од корака (енг. *Stride*), помера хоризонтално док не дође до руба, након чега се са истим кораком помера вертикално, ка доле, и враћа се на крајњу леву позицију. Поступак се понавља све док се не дође до краја матрице којом је слика представљена. Приликом сваког померања ради се матрични производ између кернела и дела матрице који је обухваћен тренутном позицијом кернела. У случају да се ради о сигналу као што је РГБ сигнал, са више канала, тада и кернел има исту дубину као и такав сигнал. Матрични производ се тада ради између сваког једнаког индекса дубине и врши се сумирање резултата са елементом помераја (енг. *bias*) како би добили једнодимензионални канал конволвираних специфичности сигнала.

Процес конволуције може се описати следећом формулом [14]:

$$x_j^i = f\left(\sum_{i \epsilon M_j} x_i^{i-1} * k_{ij}^l + b_j^l\right) \tag{3.1}$$

где је x_j^i излаз из тренутног слоја, x_i^{i-1} излаз из претходног слоја, k_{ij}^l јесте кернел за тренутни слој, b_j^l представља елементе помераја тренутног слоја, M_j представља селекцију улазног сигнала. На крају излазни сигнал пролази кроз линеарну или нелинеарну активацијску функцију (сигмоид, хиперболични тангенс, софтмакс, РЕЛУ или функције идентитета).

За активацијску функцију најчешће се бира РЕЛУ, јер у поређењу са осталима позитивно утиче на перформансе мреже. Дефинише се формулом:

$$f(x) = \max(0, x) \tag{3.2}$$

а сврха јој је да у неуронску мрежу унесе нелинеарности.

Постоје две врсте конволуције:

- Истог проширења (енг. *Same Padding*) – ако сигнал слике величине 5x5x1 проширимо, односно додамо проширење, тако да проширени сигнал има димензије 6x6x1 и применимо кернел димензија 3x3x1, добићемо излазни сигнал величине 5x5x1
- Тачног проширења (енг. *Valid Padding*) – ако на сигнал слике величине 5x5x1 применимо кернел димензија 3x3x1, добићемо излазни сигнал димензија истих као што су димензије кернела, 3x3x1

Излазна димензија O из овог слоја може се израчунати као:

$$O = \frac{W - K + 2P}{S} + 1 \tag{3.3}$$

где је O излазна ширина или висина. W представља улазну ширину/висину, K величину филтера, P проширење (енг. *Padding)*, а S корак (енг. *Stride)*. Дубина излазног сигнала зависи од броја коришћених кернела.

Број параметара p, односно елемената кернела, које неуронска мрежа треба да изучи, може се за сваки слој израчунати на основу:

$$p = (N * M * L + 1) * k \tag{3.4}$$

где N представља ширину, а M висину кернела, L дубину улазног сигнала, а k дубину кернела. Такође, за сваки индекс дубине кернела додаје се по један елемент помераја који је потребно изучити.

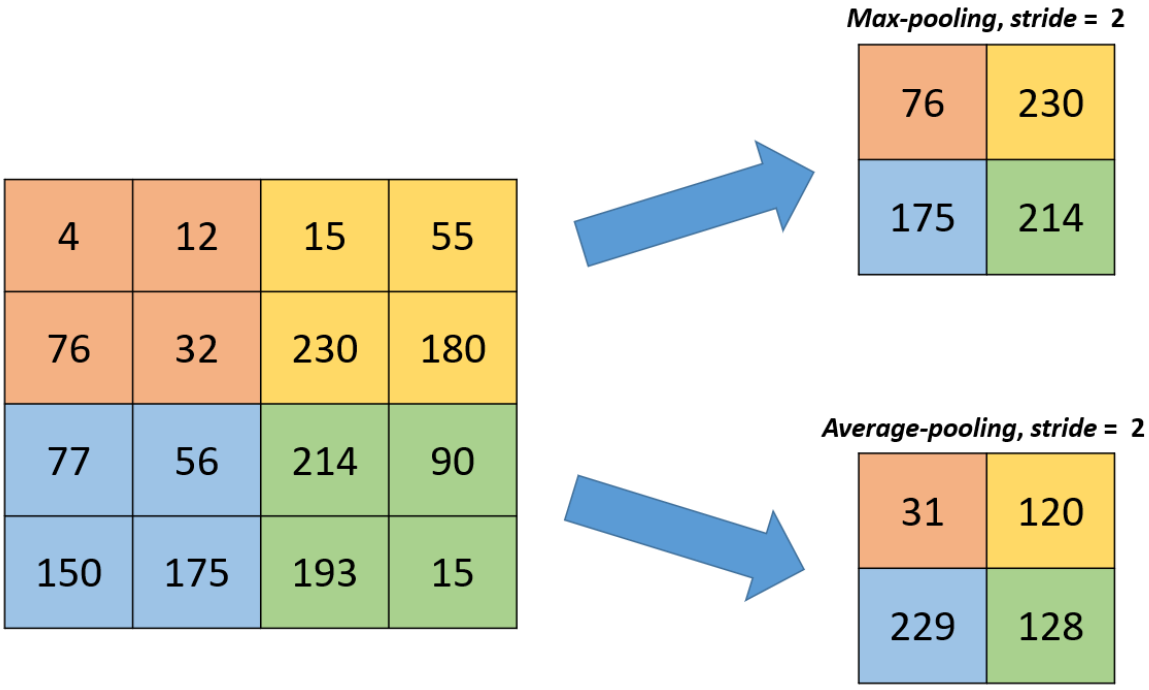
3.3.3 Сажимајући слој

Сажимајући (енг. *Pooling*) слој има могућност смањивања просторне димензије сигнала у циљу смањења снаге потребне за израчунавање и обраду података. Ако су димензије улазног сигнала $(W \times H \times C)$, димензије W (ширина) и H (висина) се мењају док димензија C (број канала или дубина) остаје непромењена.

Обично се користе један од следећа два приступа као сажимаући слој:

- *Max-pooling* – за излазну вредност узима се највећа вредности унутар дела улазног сигнала величине NxN, где N представља димензионалност кернела сажимајућег слоја
- *Average pooling* – сумира се део величине NxN улазног сигнала и проналази се средња вредност, која се узима за излазну вредност, где N представља димензионалност кернела сажимајућег слоја

Постоје и алтернативне методе сажимајућих слојева које су предложене, а неке од њих су: фракционални макс-пул (енг. *Fractional max-pool*), конволутивни пул (енг. *Sub-sampling with convolution*). [14]



Слика 3.5 Пример *Max-pool* и *Average-pool* метода са кораком = 2

Димензије излазног сигнала након примене овог слоја, могу се израчунати по формули (2.3), при чему је корак S (енг. *stride)* једнак димензији K сажимајућег слоја.

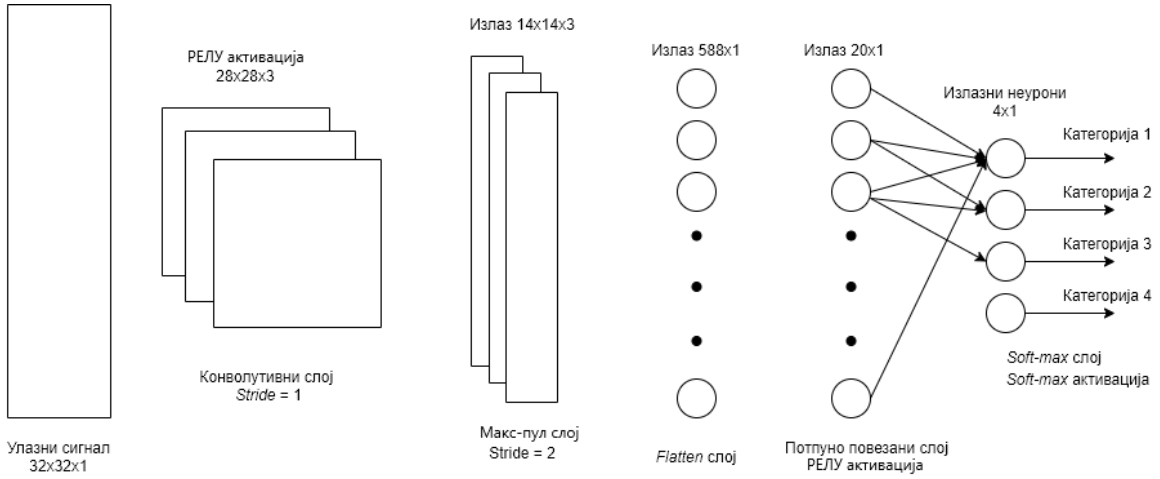
Пошто сажимајући слој узима улазни сигнал и „скупља“ га у сигнал мањих димензија, број параметара који је потребно да се изучи је једнак нули.

3.3.4 Потпуно повезани слој

Начин на који се могу научити специфичности сигнала које пролазе кроз конволутивне нивое унутар мреже јесте додавањем потпуно повезаног слоја. Излазни сигнал из конволутивних слојева, који је представљен матрицом, се исправља (енг. *Flatten*) у вектор колоне и такав пролази кроз потпуно повезани слој.

Улога потпуно повезаног слоја јесте да научи да интерпретира специфичности сигнала које долазе из конволутивних слојева и да на основу њих донесу закључак о проблему који се разматра.

Излази из потпуно повезаног слоја најчешће су вероватноће које одговарају некаквом проблему класификације, али као излаз могу се појавити и информације о позицији објеката на слици.



Слика 3.6 Потпуно повезани слој вештачких неуронских мрежа

Број тежинских коефицијената *w* овог слоја, које је потребно изучити добија се као:

$$w = (n + 1) * m \tag{3.5}$$

где n представља број улаза, а m број излаза. Такође сваком улазном тежинском коефицијенту придружује се по један елемент помераја.

3.3.5 Обучавање, параметри модела и хипер-параметри

Неуронске мреже одликују параметри и хипер-параметри модела. Обе врсте параметара су битне јер озбиљно утичу на њихове перформансе, али и на брзину обучавања. Параметри модела су они параметри које неуронска мрежа треба да научи приликом обучавања. Када причамо о конволутивним неуронским мрежама, у њих спадају параметри конволутивног слоја и тежински коефицијенти неурона у потпуно повезаним слојевима. Специфични су за модел мреже и њихове вредности се, у процесу обучавања, формирају на основу специфичности сигнала на улазу у мрежу. Хипер-параметре модела одабире руковалац неуронском мрежом и у њих спадају: функција оптимизације, активацијска функција неурона, број скривених слојева модела, итд. Хипер-параметри, као што је број скривених слојева модела, знатно утичу на рад неуронских мрежа и не постоји универзално правило по коме би се они одабрали, већ добар одабир зависи од искуства руковаоца мрежом, али и самог проблема који је потребно решити.

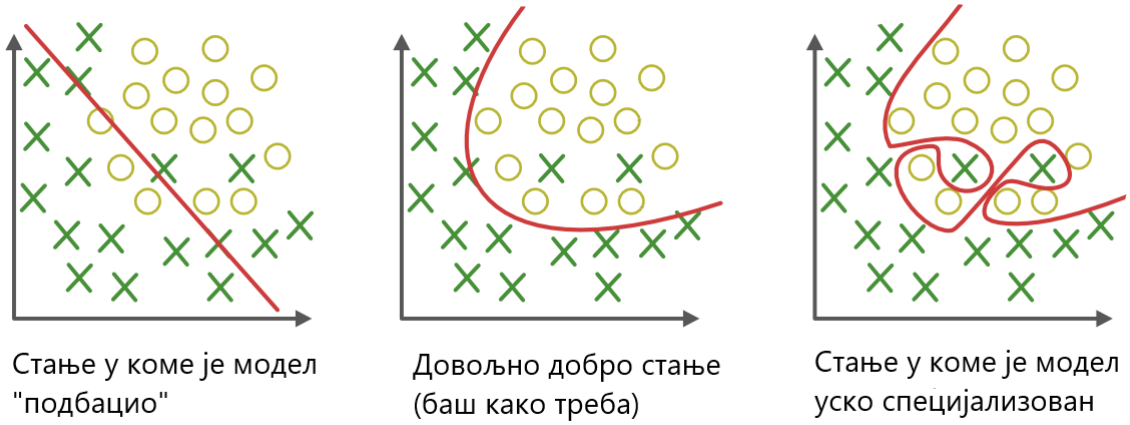
До закључка да ли су одабрани погодни хипер-параметри модела најчешће се долази уз помоћ метрике тачност валидације. Приликом анализирања метрике тачности валидације модел се може наћи у једном од три стања: стање у коме је модел „подбацио“, стање у коме је модел уско специјализован за скуп података за обучавање и довољно добро стање (баш како треба).

Уколико модел неуронске мреже „подбаци“, он неће давати жељене резултате над скупом података за обучавање, али ни над подацима које није видео приликом обучавања, односно неће имати никакву способност генерализације специфичности сигнала који је потребно обрадити. Ово стање се још назива *underfitting* и оно је нежељено стање модела неуронске мреже.

Уколико је модел неуронске мреже у стању у коме је уско специјализован за скуп података за обучавање, он ће давати изразито добре предикције над сетом података за обучавање, али ће имати јако лоше предикције над подацима које није видео приликом обучавања. Ово стање се још назива и *overfitting* и, као и претходно стање, оно је нежељено стање модела неуронске мреже.

Жељено стање у коме руковалац моделом неуронске мреже жели да се она нађе јесте довољно добро стање у коме модел може дати довољно добре резултате и над скупом података за обучавање и над подацима које није видео приликом обучавања. Да ли су

резултати довољно добри зависи од проблема који се посматра и од спремности руковаоца мрежом да толерише одређени степен грешке.



Слика 3.7 Нежељена и жељена стања модела неуронских мрежа [16]

Хипер-параметри неуронских мрежа могу се поделити на оне који су везани за структуру саме мреже и на оне који утичу на ефикасност њеног обучавања.

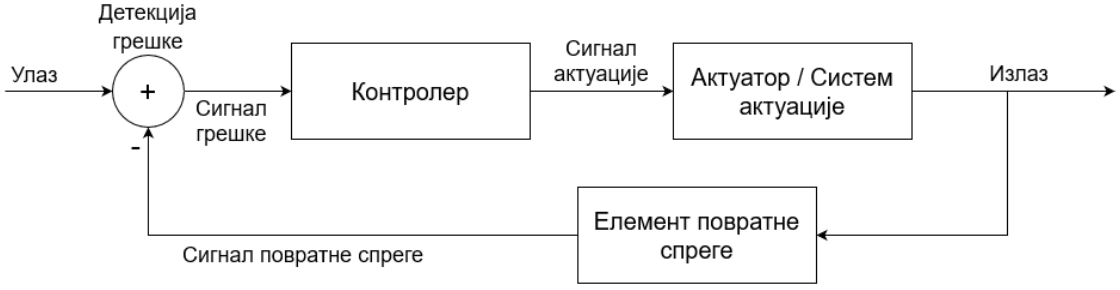
У хипер-параметре који су везани за структуру неуронске мреже спадају: број скривених слојева, иницијализација вредности тежинских коефицијената, активацијска функција и фактор одбацивања неурона приликом обучавања. Одабир броја скривених слојева неуронских мрежа зависи од комплексности проблема који се посматра и додавање додатних скривених слојева, односно мењање броја скривених слојева, може поспешити тачност рада модела. Како би неурони имали способност уочавања различитих специфичности сигнала датог проблема и да би се избегла симетрија између тежинских коефицијената различитих неурона препоручљиво је да се њихове вредности иницијализују на случајан начин. Активацијска функција директно утиче на конвергенцију неуронске мреже и на брзину обучавања и овај хипер-параметар одређује која ће се активацијска функција користити за обраду улазног сигнала унутар сваког појединачног неурона. Фактор одбацивања неурона приликом обучавања утиче на то колики постотак неурона ће бити одбачен приликом сваке епохе обучавања како би се избегло стање у коме је модел неуронске мреже „подбацио“.

У хипер-параметре који утичу на ефикасност њеног обучавања спадају: брзина обучавања (енг. *Learning rate*), смањивање брзине обучавања (енг. *Learning rate decay*), алгоритми оптимизације, број епоха обучавања и величина групе података за обучавање (енг. *Batch size*). Избор превише мале вредности хипер-параметра брзине обучавања

може довести до повећања брзине обучавања, али и машења минимума функције губитака (енг. *Loss function*) чији минимум се тражи како би се смањила грешка између предикција мреже и очекиваних резултата скупа података за обучавање. Због тога, најпогодније је увести хипер-параметар смањења брзине обучавања који ће омогућити да се смањи време обучавања како се модел мреже приближава адекватном решењу. Како би се избегло поменуто заглављивање у локалном минимуму функције губитака уводи се хипер-параметар који дефинише који ће се алгоритам оптимизације користити приликом обучавања и његов одабир зависи од проблема који се проматра. Неки од њих су: *Stohastic Gradient Descent, Momentum, Adam* и *AdaDelta,* чији описи излазе ван оквира овог рада. Број епоха обучавања говори колико пута улазни подаци за обучавање пролазе кроз неуронску мрежу приликом обучавања и пожељно га је повећавати до броја када тачност валидације над скупом података за валидацију почне да опада, па чак и ако се деси да тачност валидације над сетом података за обучавање настави да расте. Ово се ради из разлога како би се избегло нежељено стање у коме је модел неуронске мреже „подбацио“. Величина групе података за обучавање одређује колики се број података из скупа података за обучавање паралелно доводи на улаз неуронске мреже у току једне епохе јер понекад довођење целог скупа података на улаз може бити немогуће или временски скупо.

3.4 Аутоматско управљање системима

Аутоматско управљање системима бави се проучавањем аутоматских контролних система који се извршавају без непосредног деловања човека на њих. Системи аутоматског управљања састоје се од различитих елемената који врше самосталне функције. У њих се убрајају сензори, међу елементи, као и извршне јединице, односно актуатори. Аутоматски системи могу се представити структурним шемама у којима су означене везе и смерови тока сигнала, уз помоћ којих се може извршити анализа понашања система пре саме конструкције истог.



Слика 3.8 Основна контролна петља система аутоматског управљања

Детекција грешке производи сигнал грешке, који представља разлику између улазног сигнала и сигнала повратне спреге. Сигнал повратне спреге се добија из елемента повратне спреге који разматра излазни сигнал целокупног система као свој улаз. Уместо директног улаза, сигнал грешке се доводи на контролер, који производи сигнал актуације који контролише систем актуације. На основу ове логике, излазом целокупног система се контролише аутоматски док се не добије жељени одзив.

За аналитичко испитивање понашања система аутоматског управљања потребно је поставити његову диференцијалну једначину, решити је налажењем општег интеграла, одредити почетне услове и наћи зависност излазне величине од улазне уз помоћ преносне карактеристике система. Ово је често није лак проблем, посебно када је реч о сложеним системима, где се као алат за прелазак из диференцијалног у алгебарски домен једначина користе Лапласове трансформације.

Када причамо о преносној карактеристици система, она представља однос Лапласове трансформације излазног сигнала наспрам Лапалсове трансформације улазног сигнала при нултим почетним условима. На основу преносних карактеристика система, развијене су методе за испитивање разних карактеристика система, као што је стабилност која обезбеђује ограничен излазни сигнал система за ограничен улазни сигнал. Под претпоставком да елемент повратне спреге не уноси промене у излазни сигнал, преносна карактеристика система са слике 3.8 у Лапласовом домену може се записати као:

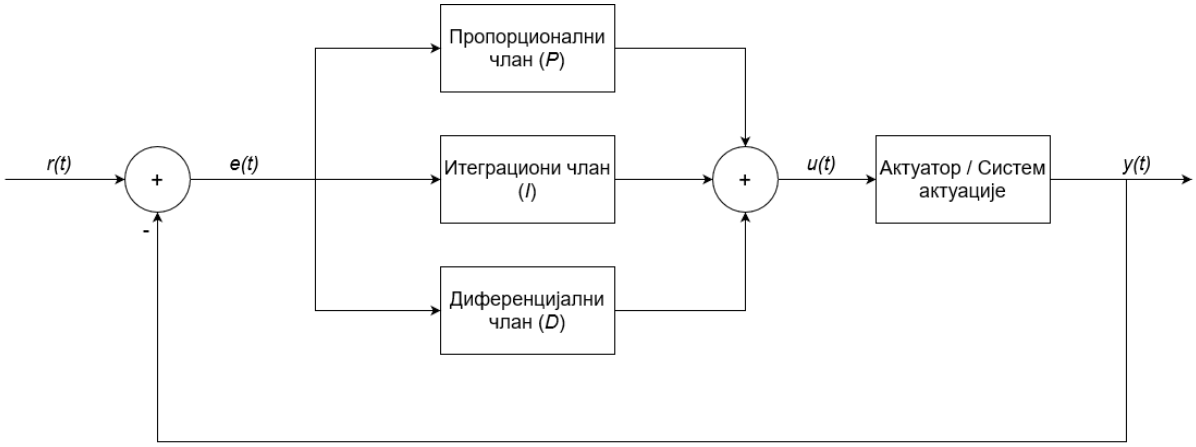
$$H(s) = \frac{K(s)G(s)}{1 + K(s)G(s)} \tag{3.6}$$

где је *K(s)* преносна карактеристика контролера, а *G(s)* преносна карактеристика система актуације. Систем ће ући у нестабилно стање када преносна карактеристика система *H(s)* дивергира за неке вредности *s*. Ово ће се десити када је *K(s)G(s)* = -1. Стабилност је загарантована када је |*K(s)G(s)*| < 1.

3.4.1 ПИД контролер

Пропорционално-интеграционо-диференцијални (ПИД) контролер је елемент контролне петље система аутоматског управљања који се користи у системима који захтевају сталну контролу излазног сигнала. ПИД контролер има могућност рачунања вредности грешке *e(t)* као разлике жељеног излазног сигнала и стварног измереног

излазног сигнала, примењујући корекцију на основу пропорционалног, интеграционог и диференцијалног члана.



Слика 3.9 Контролна петља са ПИД контролером

Контролна петља са слике 3.9 приказује принцип рада ПИД контролера, где се може приметити да ПИД контролер рачуна вредност грешке $e(t) = r(t) - y(t)$ и примењује корекцију на основу пропорционалног, интеграционог и диференцијалног члана. Уз помоћ ових чланова, ПИД контролер покушава да минимизује грешку у току времена подешавајући вредност сигнала актуације $u(t)$. Сваки од њих другачије делује на излазни сигнал:

- Пропорционални члан (P) – пропорционалан је тренутној вредности сигнала грешке $e(t)$ и у случају да је грешка велика и позитивна, сигнал актуације биће пропорционално велик и позитиван. Другим речима, повећањем пропорционалног члана повећава се брзина одзива система аутоматског управљања.
- Интеграциони члан (I) – сумира сигнал грешке $e(t)$ кроз време што резултира тиме да ће и мала грешка навести интеграциони члан да порасте са циљем што боље конвергенције излазног сигнала жељеном сигналу.
- Диференцијални члан (D) – постиже ефекат смањења сигнала актуације уколико се излазни сигнал почне интензивно повећавати. Другим речима, уколико излазни сигнал сувише брзо прилази жељној вредности сигнала, диференцијални члан деловаће ублажавајуће на сигнал актуације.

Математичка поставка ПИД контролера дата је формулом 3.7:

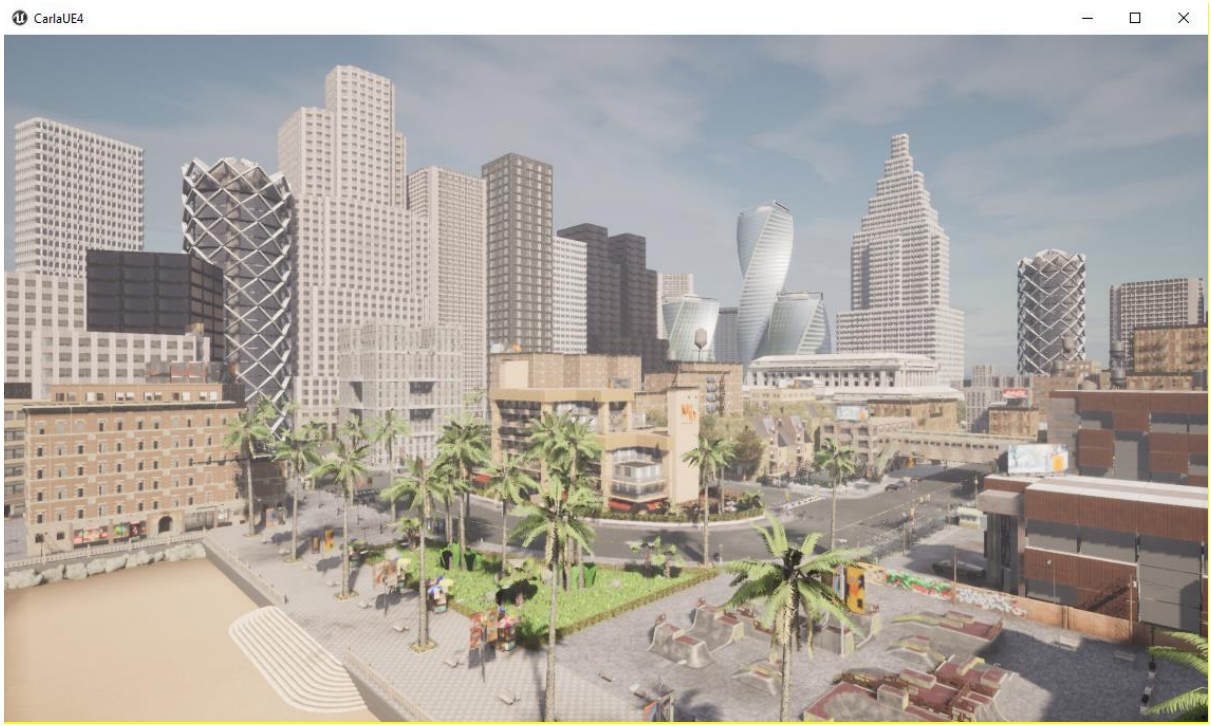
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad (3.7)$$

где K_p, K_i *и* K_d представљају коефицијенте појачања пропорционалног, интегралног и диференцијалног члана ПИД контролера, респективно. Уколико се ти фактори појачања ПИД контролера лоше изаберу, систем аутоматског управљања може постати нестабилан, односно његов излазни сигнал може дивергирати у односу на жељену вредност излазног сигнала. Поред тога, за различите проблеме могуће је искључити неки од чланова ПИД контролера, па се тако у примени могу наћи и ПИ, ПД, П или И контролери, постављајући одговарајуће коефицијенте појачања пропорционално-интеграционо-диференцијалног члана на нулу.

3.5 КАРЛА симулатор

КАРЛА симулатор представља симулатор отвореног кода направљен са циљем подржавања аспеката развоја аутономне вожње. КАРЛА пружа могућност модуларног и флексибилног корисничког интерфејса како би се олакшао развој програмских решења аутономне вожње. Сам симулатор развијен је уз помоћ *Unreal Engine 4* технологије за покретање симулација и користи последњу верзију *OpenDRIVE* стандарда за дефинисање путева и урбаних средина, која је у тренутку писања овог рада *OpenDRIVE* 1.4. [17]

Начин функционисања симулатора заснива се на клијент-сервер комуникацији, где је симулаторски сервер задужен за све радње које се тичу саме симулације, као што су: читање сензора, прерачунавање физике, ажурирање стања КАРЛА симулационог света и инстанци саобраћајних објеката у њему. Са друге стране, КАРЛА клијент састоји се од различитих модула који контролишу логику инстанци саобраћајних објеката, као и постављања поставке КАРЛА симулационог света. Контрола над симулацијама, кроз КАРЛА клијент, успоставља се уз помоћ корисничког интерфејса кроз програмски језик Питон и *C++*. [18] Поред контроле над симулацијама, КАРЛА клијент кроз исти кориснички интерфејс омогућава релативно лако добављање информација о симулацији и учесницима у саобраћају.



Слика 3.10 КАРЛА сервер

Слика 3.11 КАРЛА клијент

Како је КАРЛА симулатор сам по себи изузетно комплексан пројекат, даље у тексту дат је кратак преглед најважнијих функционалности које су се користиле приликом израде овог рада.

Неке од уграђених функционалности које овај симулатор пружа су:

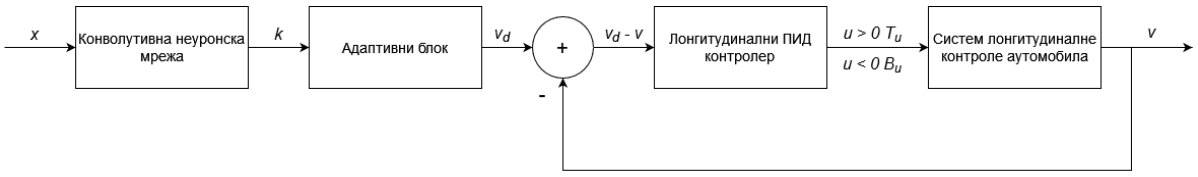
- *Управитељ саобраћајем* (енг. *Traffic manager*) – омогућава контролу свих возила, осим возила које је предмет експеримента – его возило. Служи као посредник за генерисање урбаног саобраћаја са реалистичним понашањем.
- *Сензори* – служе за прикупљање информација о окружењу у коме се возило налази, могу бити прикачени на било које возило унутар КАРЛА симулације тако да су подаци које читају доступни у било ком моменту. Неки од подржаних сензора: камера, радар, лидар, ротирајући лидар, геолокацијски сензор, и многи други.
- *Библиотека возила и окружења* – симулатор пружа разнолику библиотеку возила, почев од бицикла, преко познатих аутомобила, све до специјалних возила хитне помоћи, полицијских и ватрогасних возила. Такође, симулатор располаже сетом од 8 предефинисаних градова, са различитим саобраћајним инфраструктурама, као и различитим временским приликама.
- *Покретач сценарија* (енг. *Scenario Runner*) – омогућава покретање конструкцију и узастопно покретање различитих КАРЛА сценарија, долази као посебна библиотека програмског језика Питон.

Опис корисничког интерфејса коју пружа могућност контролисања КАРЛА симулација јавно је доступан на интернет страници КАРЛА симулатора. Приликом израде овог рада коришћена је верзија КАРЛА 0.9.12.[19] Пошто се ради о пројекту отвореног кода, код самог симулатора такође је јавно доступан, и сваком је дозвољено да допринесе пројекту или га преради за властите потребе. [20]

4 Теоријске основе програмског решења

Како би се имплементирао систем за адаптивно одржавање растојања потребно је размотрити све компоненте које су потребне за његово извршавање. Потребни елементи су: Извор фотографија – РГБ камера, механизам обраде фотографија уз могућност доношења закључака – конволутивна неуронска мрежа, компонента која поставља жељену брзину кретања возила – адаптивни блок, контролер који омогућава генерисање сигнала актуације – ПИД контролер и механизам који за улаз прима сигнал актуације – систем лонгитудиналне контроле аутомобила.

Уколико се поменуте компоненте послажу смисленим редоследом, добијамо контролну петљу овог система. Контролна петља се извршава у потпуности унутар сценарија КАРЛА симулатора и приказана је на слици 4.1.



Слика 4.1 Контролна петља система за адаптивно одржавање растојања

За реализацију система за адаптивно одржавање растојања контролну петљу потребно је рашчланити на под компоненте које чине њене градивне целине и појединачно их анализирати:

1. *Конволутивна неуронска мрежа* – прва у ланцу извршавања, даје информацију на основу улазне слике *x*, да ли се на њој налазе возила, у којој

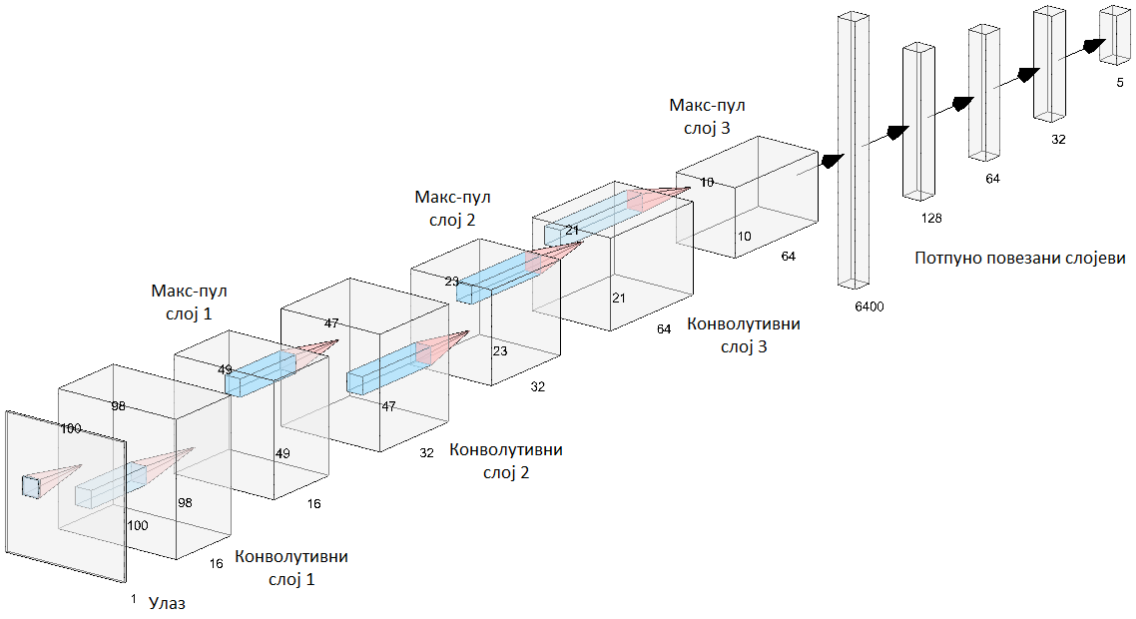
возној траци су присутна возила и колико је безбедно убрзати, да ли је потребно одржавати растојање или је потребно кочити.

2. *Адаптивни блок* – на основу излаза конволутивне неуронске мреже k поставља жељену брзину v_d и пропагира је даље у ланац извршавања.
3. *Лонгитудинални ПИД контролер* – на основу разлике између постављене жељене брзине v_d и стварне брзине v прерачунава степен актуације у виду излазног сигнала u. Уколико је сигнал u већи од нуле степен додавања гаса јесте $T_u = u$, у супротном степен кочења јесте $B_u = u$.
4. *Систем лонгитудиналне контроле аутомобила* - на основу једног од улазних сигнала T_u или B_u примењује контролу гаса или кочења.

Више информација о основним елементима контролне петље овог решења биће дато у наредним потпоглављима.

4.1 Конволутивна неуронска мрежа

Као што је споменуто, у имплементацији контролне петље овог решења система за адаптивно одржавање растојања као први градивни елемент користи се конволутивна неуронска мрежа. За потребе овог рада, конволутивна неуронска мрежа пројектована је од почетка, без наслеђених тежинских коефицијената, те обучавана над скупом података који је самостално прикупљен уз помоћ КАРЛА симулатора. Архитектура обучене неуронске мреже дата је на слици 4.2.



Слика 4.2 Архитектура конволутивне неуронске мреже

Улаз у мрежу чини црно-бела фотографија димензија 200x200 која пролази кроз први конволутивни слој чији је кернел димензија (16, 3, 3), након тога користи се макс-пул сажимајући слој димензија (2,2). Такав сигнал преузима други конволутивни слој са кернелом димензија (32, 3, 3) и поновно се користи макс-пул сажимајући слој димензија (2,2). Након тога, сигнал пролази кроз последњи конволутивни слој кернела (64, 3, 3) и користи се последњи макс-пул сажимајући слој димензија (2,2). Након тога сигнал се исправља (енг. *Flatten*) у вектор колоне и пропагира кроз три потпуно повезана слоја, при чему сваки има, респективно, 128, 64 и 32 потпуно повезана вештачка неурона. Четврти потпуно повезани слој представља излаз неуронске мреже и састоји се од 5 потпуно повезаних вештачких неурона.

4.1.1 Интерпретација излазних параметара

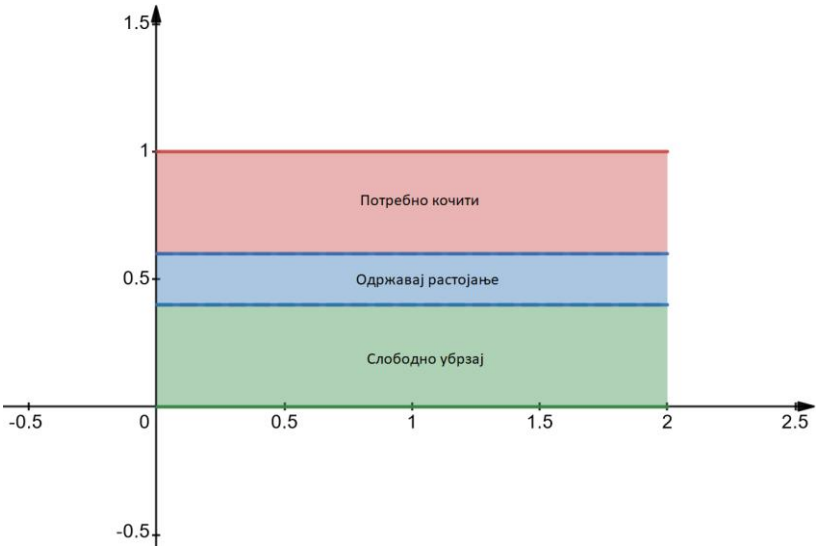
Излазне вредности неуронске мреже су у опсегу [0, 1], где прве четири вредности представљају: вероватноћу постојања возила на улазној фотографији V_p, вероватноћу постојања возила у левој суседној траци у односу на наше возило V_{pl}, вероватноћу постојања возила у десној суседној траци у односу на наше возило V_{pr}, вероватноћу постојања возила у истој траци као и наше возило V_{pc}. Последња излазна вредност интерпретира се као степен поузданости одлуке да ли је потребно убрзати или успорити наше возило или је потребно одржавати растојање између нашег возила и возила које се налази у истој траци испред нашег возила, D_{acc}. На основу тога, вектор излазних вредности, који се још назива и излазни тензор конволутивне неуронске мреже може се представити као:

$$y = [V_p\ V_{pl}\ V_{pr}\ V_{pc}\ D_{acc}] \tag{4.1}$$

Потребно је нагласити да се сви излази интерпретирају у обрнутој логици, где појављивање вредности 0 на излазима представља логичку истинитост, док појављивање вредности 1 на излазима представља логичку неистину. Поред тога, може се приметити да се на једној слици може појавити више излаза који се интерпретирају као логичка истинитост, односно на једној слици можемо имати случај да је неуронска мрежа предвидела да се возила налазе у свим суседним тракама, као и у возној траци. Оваква проблематика назива се још и проблематика вишеструких лабела (енг. *Multi-label problem*).

Уз помоћ последњег параметра D_{acc}, као што је споменуто, процењује се да ли је потребно убрзати, укочити или одржавати растојање. Како се и он креће у распону

вредности од [0, 1], тај распон вредности дели се на три дела. Распон [0, 0.4) означава да је потребно убрзати наш аутомобил. Распон [0.4, 0.6] означава да је потребно одржавати растојање од возила у истој траци и нашег возила. Распон (0.6, 1] означава да је потребно укочити наш аутомобил, као што је приказано на слици 4.3.



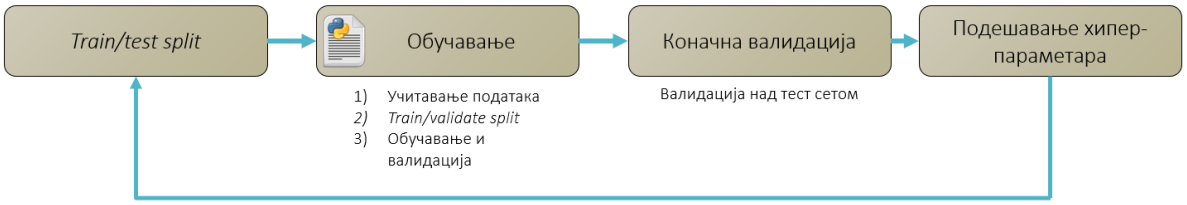
Слика 4.3 Интерпретација распона вредности параметра D_{acc}

4.1.2 Процес обучавања неуронске мреже

Прикупљање података за обучавање конволутивне неуронске мреже у виду фотографија омогућено је уз помоћ једне РГБ камере постављене на предњи део крова его возила унутар КАРЛА симулатора. Уз помоћ доступних функционалности за аутоматско кретање свих возила које пружа КАРЛА симулатор, генерише се насумичан саобраћај, те се фотографије са РГБ камере чувају на тврди диск рачунара. Овим поступком прикупљено је 16568 различитих фотографија, из различитих саобраћајних ситуација, које су ручно означене према очекиваним излазним вредностима конволутивне неуронске мреже из поглавља 4.1.1. Означене вредности похрањене су унутар *.csv* датотеке.

Као што је поменуто у поглављу 3.3.5. процес обучавања неуронских мрежа је комплексан и итеративан процес. Према томе, скуп фотографија заједно са *.csv* датотеком у којој се налазе очекиване вредности излаза неуронске мреже и који чини скуп за обучавање мреже дели се на три дела: на део који служи као тест скуп након обучавања, на део којим се мрежа директно обучава и на део који служи за валидацију модела приликом обучавања. Овај процес још се назива и обучавање/валидација/тестирање подела скупа података (енг. *train/validate/test split*). На

основу резултата над скупом за валидацију фино се подешавају хипер-параметри модела, док тест скуп служи за коначну валидацију модела и чине га подаци које мрежа никада није видела. Одређено правило по коме би се скуп података поделио на ова три дела не постоји и према томе тест скуп је величине 15 посто величине целокупног скупа, док се остатак дели на 80 посто за обучавање и 20 посто за валидацију модела.



Слика 4.4 Процес обучавања неуронске мреже

Методом насумичног одабира и праћењем метрике тачности валидације, изабрани су следећи хипер-параметри модела:

- Алгоритам оптимизације – *Stochastic Gradient Descent*
- Функција губитака – *Binary cross entropy*
- Смањивање брзине обучавања – 0.01 са стрпљењем 10 епоха
- Величина групе података за обучавање – 2
- Фактор одбацивања неурона приликом обучавања – 0.25 после сваког макс-пул сажимајућег и потпуно повезаног слоја
- Број епоха – 250 са стрпљењем између 35 и 50 епоха

Показало се, на основу [21], али и током израде овог рада, да се најбољи резултати обучавања постижу уколико се користе алгоритми оптимизације са промењивим параметром брзине обучавања.

4.1.2.1 Метрика тачности валидације

Праћењем метрике валидације процењује се успешност адаптације неуронске мреже за дати проблем, односно процењује се колико добро неуронска мрежа може да предвиди излазе за ситуације које није видела у току обучавања. С обзиром на специфичности излаза коришћене конволутивне неуронске мреже, за потребе израде овог рада развијена је сопствена метрика тачности. Прва четири излазна параметра мреже V_p, V_{pl}, V_{pr} и V_{pc} третирају као тачни уколико се вредност процентуалне разлике предикције мреже и очекивана означена вредност не разликује више од 10%. Последњи

излазни параметар D_{acc} сматра се тачним уколико се вредност процентуалне разлике предикције мреже и очекивана означена вредност не разликују више од 7%. Укупна тачност рачуна се као однос броја тачних предикција и броја укупних предикција по формули 4.2:

$$Acc = \frac{K_t}{K} \tag{4.2}$$

Где *Acc* представља тачност, K_t представља број тачних предикција, а K представља укупан број предикција.

Потреба за развијањем сопствене метрике тачности валидације произлази из проблематике вишеструких лабела, али се и емпиријски показало да се најбоља тачност постиже уколико прва четири параметра имају релаксиранији критеријум тачности од последњег параметра.

4.2 Адаптивни блок

Адаптивни блок представља други елемент контролне петље овог система за одржавање растојања и у зависности од излазних вредности конволутивне неуронске мреже диктира оптималну брзину кретања нашег возила. Другим речима, адаптивни блок анализира резултате обраде конволутивне неуронске мреже, те на основу одређених услова и анализе поставља брзину којом се гарантује безбедна реакција остатка система на возно окружење.

Уколико се анализом првог излазног параметра неуронске мреже V_p закључи да на улазној фотографији не постоје возила у окружењу нашег возила, жељена брзина v_d поставља се на максималну дозвољену брзину за тип пута по којем се вози. У супротном, уколико на улазној фотографији постоји бар једно возило, улази се у дубљу анализу излазних параметара неуронске мреже. У оквиру дубље анализе, проматра се последњи излазни параметар неуронске мреже D_{acc}, те уколико је вредност тог параметра у опсегу [0, 0.4), адаптивни блок поставља жељену брзину v_d на максималну дозвољену брзину за тип пута по којем се вози. Како вредност параметра D_{acc} расте, и улази у распон вредности [0.4, 0.6] адаптивни блок смањује жељену брзину кретања нашег возила како би наше возило прилагодило своју брзину кретању возила које се налази испред њега и тако одржавало растојање. Уколико се вредност параметра D_{acc} нађе у опсегу вредности (0.6, 1] адаптивни блок ће поставити жељену брзину кретања нашег возила на нула километара на час, све док се вредност параметра D_{acc} поново не нађе у опсегу од [0.4,

0.6] када ће се жељена брзина кретања v_d поставити на вредност која је претходно сачувана пре него је параметар D_{acc} прешао из опсега [0.4, 0.6] у опсег (0.6, 1]. Интуитивно, уколико параметар D_{acc} пређе из опсега [0.4, 0.6] у опсег [0, 0.4), адаптивни блок за жељену брзину v_d поновно поставља максималну дозвољену брзину за тип пута по којем се вози.

4.3 Лонгитудинални ПИД контролер

Као што је већ поменуто у поглављу 3.4.1, ПИД контролер прерачунава вредност грешке прорачунавајући вредност разлике између постављене жељене вредности и измерене вредности процеса и примењујући корекцију уз помоћ пропорционалне, интеграционе и диференцијалне компоненте. У нашем случају, жељена вредност процеса биће жељена брзина кретања аутомобила v_d, а измерена вредност процеса тренутна вредност брзине кретања нашег аутомобила v. Посматрајући формулу 4.3, излазни сигнал ПИД контролера можемо описати као:

$$u = K_p(v_d - v) + K_i \int_0^t (v_d - v)\, dt + K_d \frac{d(v_d - v)}{dt} \qquad (4.3)$$

са T_u означићемо степен притискања папучице гаса, а са B_u степен притискања папучице кочнице. Уколико је $u \geq 0$, $T_u = u$, $B_u = 0$, у супротном уколико је $u < 0$, $T_u = 0$, $B_u = -u$.

Како би се избегло нагло додавање гаса или нагло притискање кочнице, минимизација разлике између жељене брзине v_d и стварне брзине возила v, не сме бити у великим корацима. Другим речима, стварна брзина возила v треба да конвергира у мањим корацима до жељене брзине v_d како се путници у аутомобилу не би осећали нелагодно и како би се минимизовало трошење механичких делова аутомобила. Ово се постиже наштимавањем коефицијената пропорционалне, интеграционе и диференцијалне компоненте, респективно K_p, K_i и K_d.

4.3.1 Наштимавање коефицијената K_p, K_i и K_d

Иако постоје разне методе за проналажење оптималног решења задатог проблема, за наштимавање коефицијената пропорционалне, интеграционе и диференцијалне компоненте ПИД контролера коришћеног у овом систему искоришћена је метода претраге мреже (енг. *Grid search*). Метода претраге мреже одликује се проласком кроз све могуће наведене комбинације тражених параметара примењујући их итеративно унутар једне симулације понашања система.

Приликом наштимавања коефицијената пропорционалне, интеграционе и диференцијалне компоненте за кориштени ПИД контролер, могуће вредности коефицијената кориштене су вредности у распону од 10^{-5} до 10^{5} са фактором множења 10:

$$K_p = [10^{-5}\ 10^{-4}\ 10^{-3}\ 10^{-2}\ 10^{-1}\ 10^{0}\ 10^{1}\ 10^{2}\ 10^{3}\ 10^{4}\ 10^{5}]$$

$$K_i = [10^{-5}\ 10^{-4}\ 10^{-3}\ 10^{-2}\ 10^{-1}\ 10^{0}\ 10^{1}\ 10^{2}\ 10^{3}\ 10^{4}\ 10^{5}]$$

$$K_d = [10^{-5}\ 10^{-4}\ 10^{-3}\ 10^{-2}\ 10^{-1}\ 10^{0}\ 10^{1}\ 10^{2}\ 10^{3}\ 10^{4}\ 10^{5}] \qquad (4.4)$$

Проласком кроз тродимензионалну мрежу коефицијената, за сваку итерацију чувају се подаци о излазном сигналу из ПИД контролера, те се функцијом доброте одређује ефикасност понашања ПИД контролера за сваку комбинацију коефицијената K_p, K_i и K_d.

Како наредни елемент контролне петље овог решења који делује на механичке делове аутомобила, о коме ће бити више речи у поглављу 4.4, прима сигнале у распону вредности [0, 1], а чије су улазне вредности заправо излазне вредности ПИД контролера, функција доброте омогућава њихову оптимизацију. Како не би долазило до наглог убрзавања и пренаглог кочења, излазни сигнали ПИД контролера ограничени су у распону вредности [-0.7, 0.7]. Пренагло кочење или убрзавање аутомобила, поред утицаја на удобност вожње може довести и до проклизавања аутомобила у оба случаја, а како је КАРЛА симулатор користи реалне моделе аутомобила, ова појава се огледа и у њему.

Функција доброте прво елиминише све нулте вредности излазног сигнала ПИД контролера, те рачна број вредности које упадају у поменути интервал. Напослетку, рачуна се однос броја вредности које упадају у интервал [-0.7, 0.7] и број укупних ненултих излазних вредности ПИД контролера, по формули 4.5:

$$f(u) = \frac{u}{u_t} \qquad (4.5)$$

Где *f(u)* представља доброту излазног сигнала ПИД контролера и што је тај број ближе јединици, одзив ПИД контролера се сматра бољим, *u* представља број вредности које упадају у интервал [-0.7, 0.7], а u_t представља укупан број ненултих излазних вредности ПИД контролера.

Након иницијалног проласка кроз тродимензионалну мрежу коефицијената, врши се доштимавање коефицијената поновним проласком кроз нову тродимензионалну

мрежу чије се вредности налазе у околини вредности које применом функције доброте дају најбоље резултате.

Применом методе претраге мреже и коришћењем реализоване функције доброте најбоље понашање ПИД контролера остварено је са коефицијентима из табеле 4.1.

Коефицијент	Вредност
K_p	1.0
K_i	0.8
K_d	0.1

Табела 4.1 Наштимани коефицијенти ПИД регулатора

Због природе методе претраге мреже, време потребно за проналажење оптималних параметара у најгорем случају расте експоненцијално са величином мреже. Време извршавања може се израчунати према формули 4.6:

$$t = (n_1 * n_2 \ldots * n_n) * t_s \tag{4.6}$$

где n_1, n_2, … n_n, представља број параметара за претрагу по оси мреже претраге, а t_s представља просечно време извршавања једне симулације.

У случају наштимавања коефицијената коришћеног ПИД контролера, за по 11 параметара на свакој од три осе мреже претраге и просечног времена извршавања једне симулације које износи 50 секунди, укупно време за проналазак оптималних K_p, K_i и K_d коефицијената износи 18.48 часова.

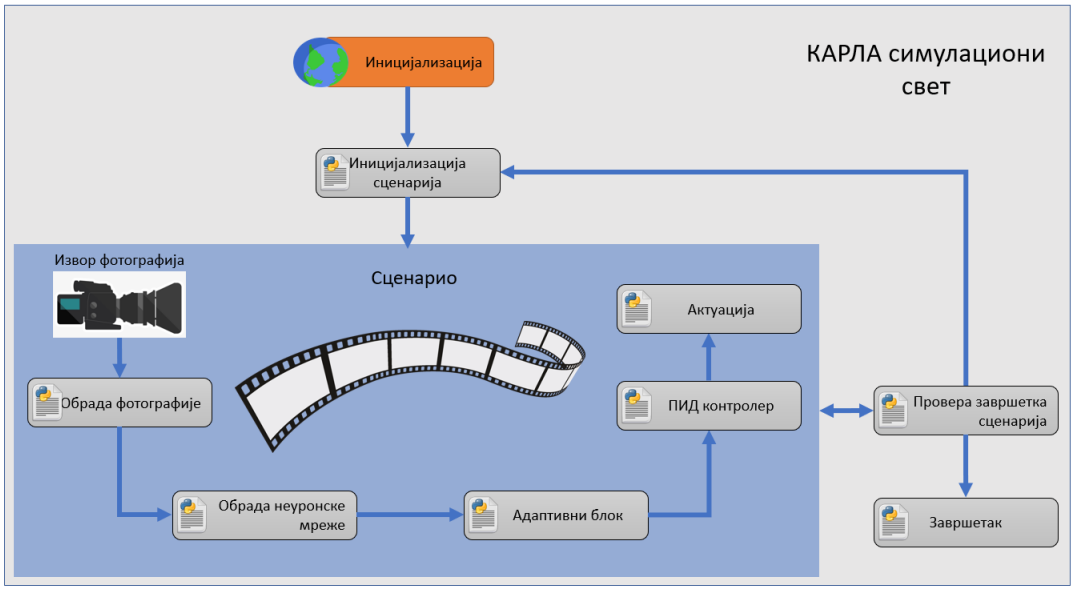
4.4 Систем лонгитудиналне контроле аутомобила

Систем лонгитудиналне контроле аутомобила подразумева систем који управља командама аутомобила за убрзавање и успоравање – папучицом гаса, као и папучицом кочнице. Начин управљања овим командама зависи од много фактора и специфичности самог возила којим се управља. У неке од тих специфичности спадају, када се ради о додавању гаса и тип мотора који је уграђен у аутомобил, врста горива која се убризгава у сам мотор, као и електроничке контролне јединице (енг. *ECU*) која надгледа убризгавање горива у мотор. Када је реч о кочењу неки од фактора који утичу на њега јесу: брзина кретања аутомобила, тежина аутомобила, материјали од којих је изграђен кочиони систем, трење између точкова и пута, итд. Обједињено, све акције које утичу на кретање аутомобила зависе од динамичког модела аутомобила који описује његово кретање у простору, чије пројектовање превазилази оквире овог рада.

С обзиром да је имплементација овог система реализована унутар КАРЛА симулатора, искоришћена је већ постојећа имплементација система лонгитудиналне контроле аутомобила која прима вредности у распону [0, 1] и када је реч о додавању гаса и када је реч о стискању кочнице.

5 Програмско решење

У овом поглављу описано је програмско решење система за адаптивно одржавање растојања. За потребе овог рада, као што је већ поменуто, искоришћен је симулатор отвореног кода КАРЛА, који нам релативно лако обезбеђује потребне функционалности за генерисање саобраћаја, његово кретање у симулационом простору и прилагођавање корисниковим потребама кроз кориснички интерфејс Питон програмског језика.



Слика 5.1 Ток података унутар КАРЛА симулационог света

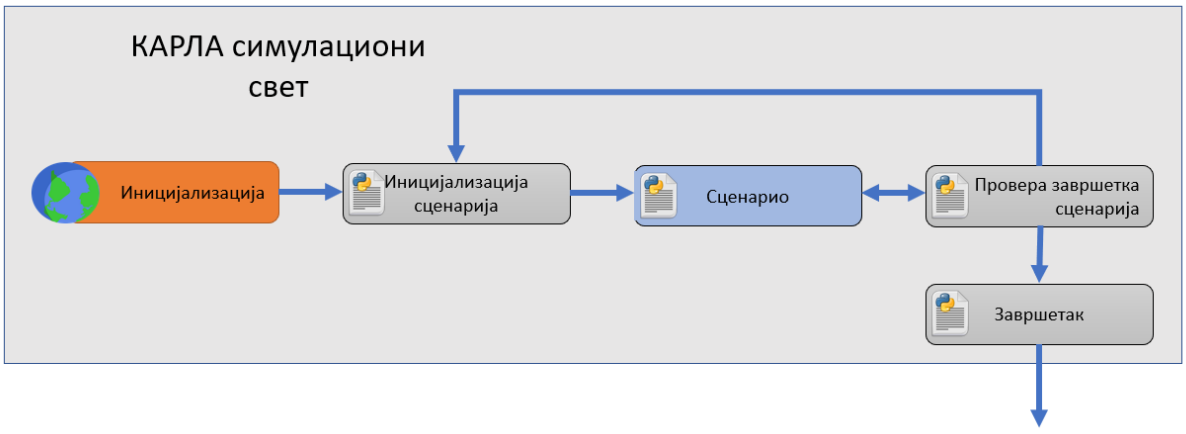
На основу тока података унутар КАРЛА симулационог света са слике 5.1, програмско решење можемо посматрати са два нивоа апстракције:

1. Постојање КАРЛА симулационог света унутар којег се извршавају различити сценарији
2. Постојање различитих сценарија унутар КАРЛА симулационог света који је садрже имплементацију овог решења система за адаптивно одржавање растојања.

Према томе, наредна потпоглавља описују програмско решење са оба нивоа апстракције референцирајући се на слику 5.1.

5.1 КАРЛА симулациони свет

КАРЛА симулатор нам, као што је већ поменуто, омогућава креирање КАРЛА симулационог света у коме се може креирати реалан сценарио жељених саобраћајних ситуација. Оно што се може сматрати као недостатак КАРЛА симулатора јесте што не постоји, поред споменуте додатне библиотеке покретача сценарија, уграђени механизам који би омогућио креирање више различитих сценарија и њихово секвенцијално извршавање. Међутим, приликом израде овог рада примећено је и да је поменута библиотека изразито тешка за постављање, комплексна и нестабилна. Због тога за потребе овог рада, у програмском језику Питон имплементиран је један такав механизам који омогућава поменуту функционалност. Симболично, овај механизам назван је *SimScenarioRunner*, и имплементира га истоимена класа програмског језика Питон.



Слика 5.2 Ток података унутар КАРЛА симулационог света 2

Са слике 5.2 да се приметити неколико блокова: „Иницијализација“, „Иницијализација сценарија“, „Сценарио“, „Провера завршетка сценарија“ и „Завршетак“. Блок „Иницијализација“ задужен је за добављање информација о КАРЛА симулационом свету који се у суштини иницијализује у оквиру КАРЛА сервера, који није предмет обраде овог рада. Овај блок служи као улазна тачка у симулационо окружење. Блок „Сценарио“ предмет је обраде наредног потпоглавља 5.2. Блокови „Иницијализација сценарија“, „Провера завршетка сценарија“ и „Завршетак“ део су *SimScenarioRunner* механизма и служе како би се надоместио поменути недостатак КАРЛА симулатора.

Приликом конструкције *SimScenarioRunner* објекта кроз програмски језик Питон, као параметри објекта шаљу се информације о клијенту КАРЛА симулатора (који садржи информације о КАРЛА симулационом свету), услове за завршетак сваког појединачног сценарија, те за сваки сценарио почетне позиције околног саобраћаја и почетна позиција его возила којим систем за адаптивно прилагођавање брзине управља. Подаци о саобраћају представљени су матрицама у којима једна врста представља један сценарио, а потребни су *SimScenarioRunner* објекту како би успешно могао да затражи од КАРЛА сервера креирање одговарајућих саобраћајних учесника кроз блок „Иницијализација сценарија“.

Блок „Иницијализација сценарија“ имплементира методу *initScenario* која као параметар прима индекс сценарија који идући треба да се иницијализује, проверава да ли је потребно деиницијализовати претходни сценарио, те иницијализује нови. Имеђу осталог, ова метода омогућава поставља потребан РГБ камера сензор на его возило, те ПИД контролер. По извршавању овог блока, прелази се на извршавање сценарија кроз блок „Сценарио“. Након што се изврши један симулациони корак сценарија, *SimScenarioRunner* механизам кроз блок „Провера завршетка сценарија“ имплементира методу *nextScenario* која проверава да ли су се стекли услови за завршетак тренутног сценарија. Уколико нису, контрола се назад препушта блоку „Сценарио“, те се овај поступак понавља све док се не стекну услови потребни за његово завршавање. Када се стекну ти услови, проверава се да ли је то последњи сценарио у листи сценарија које је потребно извршити. Уколико јесте, контрола тока се препушта блоку „Завршетак“ који шаље поруку КАРЛА серверу да је потребно деиницијализовати све објекте који су генерисани приликом извршавања последњег блока „Иницијализација сценарија“, те да КАРЛА клијент завршава са радом. Уколико су се стекли услови за завршетак сценарија, а то није последњи сценарио у листи сценарија које је потребно извршити, контрола тока

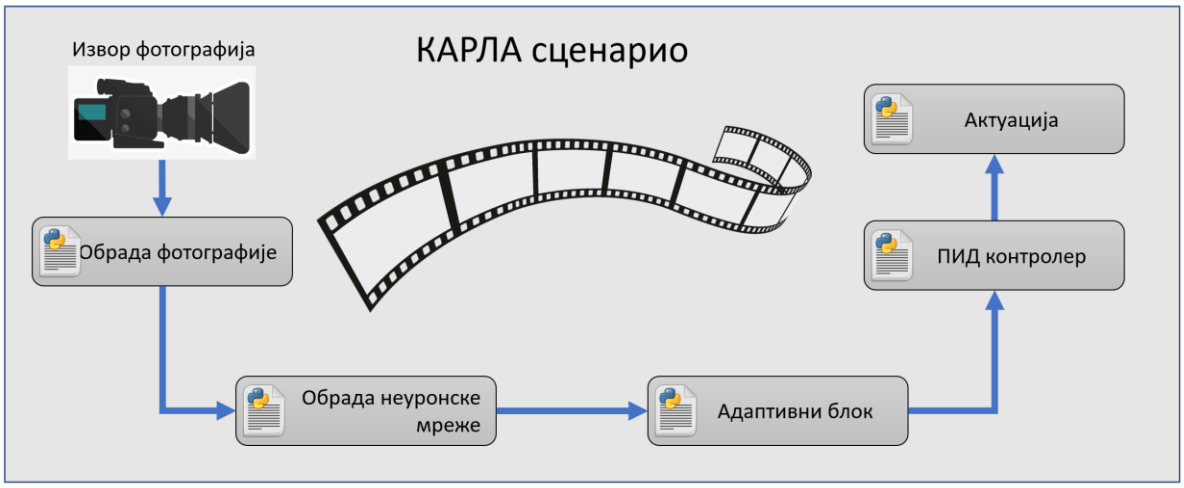
се препушта блоку „Иницијализација сценарија“ све док се не стекну услови препуштање контроле блоку „Завршетак“.

Поред описаних функционалности, *SimScenarioRunner* имплементира методу *periodicSimScenarioRunner* која се може користити за увођење произвољних динамичких промена сценарија у току времена извршавања (форсирање одређеног возила на промену возне траке, интеракција са саобраћајном инфраструктуром у тачно дефинисаном тренутку, итд.).

Како КАРЛА симулатор функционише по принципу сервер-клијент комуникације, мана овако имплементираног механизма јесте што је поменута компонента имплементирана са клијентске стране. Имплементација овог механизма унутар серверске компоненте КАРЛА симулатора, пружајући кориснички интерфејс ка клијентској компоненти, убрзала би извршавање узастопних иницијализација, деиницијализација и креирања саобраћајних учесника и објеката.

5.2 КАРЛА сценарио

Када *SimScenarioRunner* механизам иницијализује све потребне услове за извршавање једног сценарија у оквиру КАРЛА симулационог света контрола тока података прелази у његово извршавање. Ток података унутар КАРЛА сценарија за ово решење система за адаптивно одржавање растојања приказана је на слици 5.3.



Слика 5.3 Ток података унутар КАРЛА сценарија

На его возило којим управљамо прикачени су РГБ камера и ПИД контролер који омогућавају реализацију контролне петље система за адаптивно одржавање растојања са слике 5.3. РГБ камера у овом случају служи као извор фотографија димензија 1280x720 пиксела. Када РГБ камера забележи фотографију прослеђује је блоку „Обрада

фотографије“ који је припрема за блок „Обрада неуронске мреже“. Припрема фотографије за обраду на конволутивној неуронској мрежи подразумева исецање регије од интереса са целокупне фотографије, како би се избацили ирелевантни делови који не уносе нове информације у систем, а додатно га оптерећују својом димензионалношћу. Регија од интереса димензија 600x370 пиксела приказана је на слици 5.4. Поред исецања регије од интереса, таква исечена слика се из РГБ простора боја претвара у црно-белу фотографију, скалира на димензије улазног слоја конволутивне неуронске мреже димензија 200x200 пиксела, те нормализује у опсег вредности [0, 1].

Слика 5.4 Регија од интереса конволутивне неуронске мреже

Над таквом пред обрађеном фотографијом, конволутивна неуронска мрежа врши предикције над чијим се излазним параметрима из поглавља 4.1.1 врши примена технике која ублажава грешке предикција неуронске мреже. Наиме, како неуронска мрежа уме својевремено да погреши у давању предикција, како би се избегло нагло деловање система, или чак погрешна одлука адаптивног блока, примењује се прозор предикција константне величине над којим се рачуна његова средња вредност. Ова техника још се назива и *Simple Moving Average*. Поступак је итеративан и прозор се помера у складу са пристизањем нових предикција (A_n), тако да се средња вредност рачуна над n најмлађих предикција, по формули 5.1.

$$SMA = \frac{A_1 + A_2 + \cdots + A_n}{n}, \tag{5.1}$$

Након примене *Simple Moving Average* технике на предикције конволутивне неуронске мреже, оне се пропагирају у наредни блок обраде „Адаптивни блок“ који има улогу да постави жељену брзину кретања возила. „Адаптивни блок“ проверава вредности излаза неуронске мреже и одлучује о наредним акцијама које је потребно предузети. Уколико неуронска мрежа закључи са поузданошћу већом од 50 % да се на улазној фотографији не налазе возила, „Адаптивни блок“ поставља максималну дозвољену брзину за тип пута по којем се вози као жељену брзину, у супротном улази се у дубљу анализу параметара како би се донела одлука жељене брзине. Уколико последњи параметар неуронске мреже који говори о степену поузданости убрзавања/одржавања растојања/кочења покаже да је могуће убрзати, односно налази се у распону вредности [0, 0.4) „Адаптивни блок“ поставља максималну дозвољену брзину за тип пута по којем се вози као жељену брзину. У супротном, уколико је вредност тог параметра у распону вредности (0.6, 1] као жељена брзина поставља се 0 као индикација да је его возило сувише близу возила испред себе те је потребно кочити. Уколико се последњи параметар неуронске мреже нађе у распону вредности [0.4, 0.6] „Адаптивни блок“ уз помоћ спремника претходних вредности проверава стопу промене вредности овог параметра кроз неколико сукцесивних предикција. Стопа промене вредности даје информацију о расту или паду одређене промењиве, те њена позитивна вредност означава да промењива расте кроз време, док у супротном, негативна вредност означава пад промењиве кроз време. Ово се може описати формулом 5.2:

$$ROC = (\frac{x_{t-1}}{x_t} - 1) * 100 \qquad (5.2)$$

где x_{t-1} представља претходну вредност промењиве у временском тренутку $t-1$, а x_t представља тренутну вредност промењиве у временском тренутку t.

У тренутку када се уђе у стање у коме је потребно одржавати растојање, „Адаптивни блок“ памти тренутну брзину возила као жељену брзину кретања.Уколико је вредност стопе промене позитивна „Адаптивни блок“ ће умањити жељену брзину кретања 7.5 километара на час од тренутно постављене жељене брзине , а у супротном уколико је она негативна повећаће тренутно постављену жељену брзину за 7.5 километара на час. „Адаптивни блок“ је имплементиран на начин да се у стању одржавања растојања брзина не може смањивати на мање од половине брзине којом се возило кретало пре него је установљено да је потребно одржавати растојање. Такође

жељена брзина се не може повећавати више од максималне дозвољене брзине за тип пута по којем се вози.

Након адаптивног блока ток података се креће до ПИД контролера које је везано на његово возило и коме се прослеђује утврђена жељена брзина кретања. На основу тренутне брзине, жељене брзине и пропорционално-интеграционо-диференцијалних коефицијената ПИД контролера прилагођава се сигнал актуације који утиче на степен притискања гаса, односно уколико је то потребно, степен притискања папучице кочнице. Блок „Актуација“ саставни је део функционалности КАРЛА симулатора који за параметре прима горе поменути сигнал актуације, чиме се постиже деловање на аутомобил и промену његовог кретања.

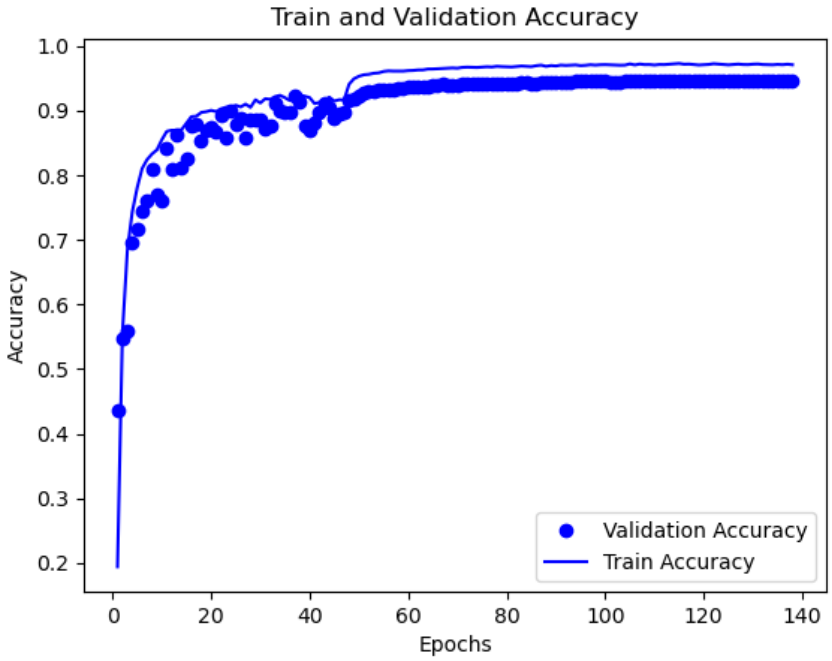
Описани процеси понављају се док се не стекну услови за завршетак сценарија.

6 Резултати

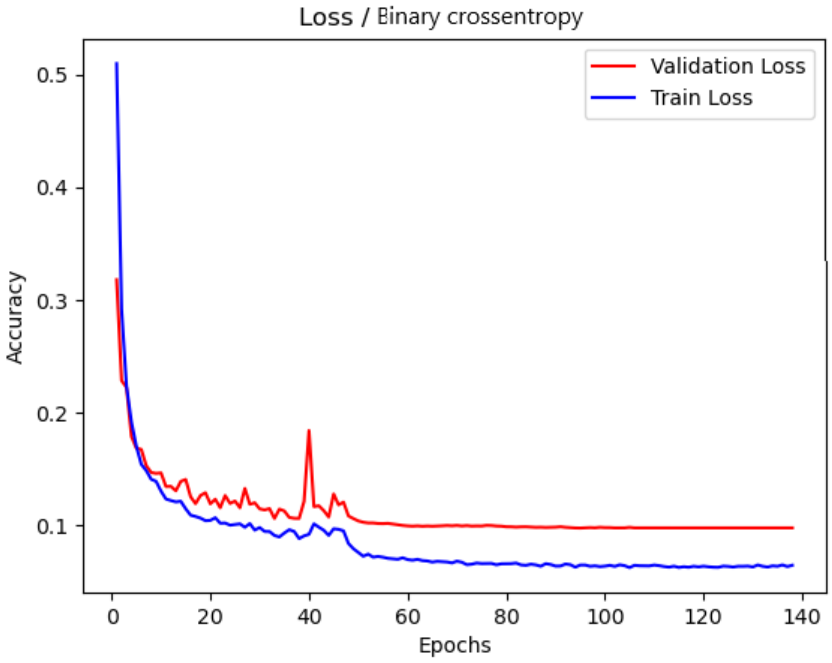
У овом поглављу дат је осврт на постигнуте резултате обучавања конволутивне неуронске мреже и њене примене кроз различите саобраћајне ситуације унутар КАРЛА симулатора.

6.1 Резултати обучавања конволутивне неуронске мреже

На графиконима 6.1 и 6.2 приказани су тачности обучавања и валидације модела, као и губици током процеса обучавања.



Графикон 6.1 Резултати тачности обучавања и валидације модела конволутивне неуронске мреже



Графикон 6.2 Губици приликом обучавања конволутивне неуронске мреже

Са графикона 6.1 може се видети да је постигнута тачност обучавања 0.9729, односно 97.29% док је постигнута тачност обучавања приликом валидације над подацима које конволутивна неуронска мрежа није видела износи 0.9461, односно

94.61%. С обзиром на величину скупа података, од 16568 различитих фотографија, постигнута тачност валидације приликом обучавања за формирани проблем, може се сматрати високом и задовољавајућом за потребе овог рада.

Са друге стране, на графикону 6.2 који приказује губитке приликом обучавања, губици приликом обучавања минимизовани су на 0.0629, док су губици приликом валидације над подацима које неуронска мрежа није видела јесте 0.0979. Разматрајући ове бројке, али и изглед самог графикона губитака, може се закључити да је неуронска мрежа успешно савладала формирани проблем, без одласка у нежељена стања у коме је модел „подбацио“ нити у стање у коме је уско специјализован за скуп података за обучавање.

6.2 Резултати система за адаптивно одржавање растојања

У наредним потпоглављима разматрани су различите саобраћајне ситуације формиране кроз КАРЛА симулатор.

6.2.1 Први симулациони случај – празна цеста

У овом симулационом случају его возило се вози по празној отвореној цести.

Слика 6.1 Симулациони случај – празна цеста

Резултати симулације показују да неуронска мрежа не препознаје нити једно возило на цести у било којој траци и последњи параметар D_{acc} поуздано говори да се

може убрзати. Ови закључци неуронске мреже говоре адаптивном блоку да слободно може да жељена брзина кретања его возила може слободно поставити на максималну дозвољену брзину за тип цесте по којој се вози, док ПИД контролер обезбеђује сигнал актуације. У овом случају максимална дозвољена брзина поставља се на 80 километара на час.

6.2.2 Други симулациони случај – текући саобраћај (са водећим возилом)

У овом симулационом случају его возило се вози по отвореној цести са свих страна окружено возилима и са возилом испред себе у траци у којој се вози.

Слика 6.2 Симулациони случај – текући саобраћај, са водећим возилом

Резултати симулације показују да неуронска мрежа детектује возила на слици, возила у свакој траци и у зависности од ситуације уз помоћ параметра D_{acc} говори да ли се може убрзати, да ли је потребно држати одстојање од водећег возила или је потребно кочити. У зависности од тога адаптивни блок поставља жељену брзину кретања, а ПИД регулатор обезбеђује сигнал актуације како би се постигла жељена брзина кретања. Експеримент се извршава успешно, без наглог кочења, наглог заустављана или колизије са водећим возилом.

6.2.3 Трећи симулациони случај – текући саобраћај (без водећег возила)

У овом симулационом случају его возило се вози по отвореној цести са свих страна окружено возилима и без возила испред себе у траци у којој се вози.

Слика 6.3 Симулациони случај – текући саобраћај, без водећег возила

Резултати симулације показују да неуронска мрежа детектује возила на слици, возила у свакој траци, али не и у траци којом се его возило креће и уз помоћ параметра D_{acc} говори да се може убрзати. Ови закључци неуронске мреже говоре адаптивном блоку да слободно може да жељена брзина кретања его возила може слободно поставити на максималну дозвољену брзину за тип цесте по којој се вози, док ПИД контролер обезбеђује сигнал актуације. У овом случају максимална дозвољена брзина поставља се на 80 километара на час. Експеримент се извршава успешно, без наглог кочења или наглог заустављана.

6.2.4 Четврти симулациони случај – возило се престројава у траку его возила

У овом симулационом случају его возило се вози по отвореној цести када се у једном тренутку возило из суседне леве траке нормално престројава у возну траку его возила.

Слика 6.4 Симулациони случај – возило се престројава у траку его возила

Резултати симулације показују да неуронска мрежа детектује возила у суседној траци, али не и у возној траци којом се его возило креће, до тада параметар D_{acc} указује да се слободно може убрзати. Што адаптивни блок и ПИД контролер поштују постављањем жељене брзине кретања на максималну дозвољену брзину кретања за тип пута по којем се креће и добрим сигналом актуације. У овом случају максимална дозвољена брзина поставља се на 80 километара на час. У моменту када возило из суседне леве траке крене да се престројава у траку его возила, неуронска мрежа детектује возило и у левој траци и у возној траци его возила, док год се возило у потпуности не престрои, када га детектује само у возној траци его возила, али не и у левој траци. D_{acc} се повећава и диктира да је потребно одржавати растојање од возила испред, што се адаптивним блоком и сигналом актуације ПИД контролера одражава на брзину кретања его возила. Експеримент се завршава успешно без колизије са возилом које се престројило у возну траку его возила.

6.2.5 Пети симулациони случај – возило се нагло престројава у траку его возила

У овом симулаиционом случају его возило се вози по отвореној цести када се у једном тренутку возило из суседне десне траке нагло престројава у возну траку его возила и одсеца га.

Слика 6.5 Симулациони случај – возило се нагло престројава у траку его возила

Резултати симулације показују да неуронска мрежа детектује возила у суседној траци, али не и у возној траци којом се его возило креће, до тада параметар D_{acc} указује да се слободно може убрзати. Што адаптивни блок и ПИД контролер поштују постављањем жељене брзине кретања на максималну дозвољену брзину кретања за тип пута по којем се креће и добрим сигналом актуације. У овом случају максимална дозвољена брзина поставља се на 80 километара на час. У моменту када возило из суседне леве траке крене да се престројава у траку его возила, неуронска мрежа детектује возило и у десној траци и у возној траци его возила, док год се возило у потпуности не престрои, када га детектује само у возној траци его возила, али не и у левој траци. D_{acc} се нагло повећава и диктира да је потребно нагло закочити. Када возило које се престројило крене да се удаљава од нашег возила D_{acc} се смањује до момента када је поново гарантовано сигурно кретање его возила. Експеримент се завршава успешно без колизије са возилом које се престројило у возну траку его возила.

6.2.6 Шести симулациони случај – его возило наилази на статички саобраћај

У овом симулационом случају его возило се креће по отвореној цести до момента када наилази на саобраћај који је у застоју.

Слика 6.6 Симулациони случај – статички саобраћај

Резултати симулације показују да неуронска мрежа не детектује возила нити у једној траци делом пута где он не постоји, а параметар D_{acc} указује да се може убрзавати. Што адаптивни блок и ПИД контролер поштују постављањем жељене брзине кретања на максималну дозвољену брзину кретања за тип пута по којем се креће и добрим сигналом актуације. У овом случају максимална дозвољена брзина поставља се на 80 километара на час. У моменту када се појављује статички саобраћај, неуронска мрежа почиње да детектује возила у свим возним тракама, а параметар D_{acc} полако расте и указује да је потребно кочити. Адаптивни блок и ПИД контролер поштују постављањем жељене брзине кретања на нула километара на час и добрим сигналом актуације. Експеримент се завршава успешно без колизије са статичким саобраћајем, праводобним и угодним кочењем по путнике его возила.

7 Закључак

У овом праду описано је једно решење система за адаптивно одржавање растојања које за своје градивне целине користи принципе машинског учења и конволутивних неуронских мрежа као механизма за обраду фотографије, али и принципе система аутоматског управљања како би се деловало на механичке делове аутомобила. Иако изразито осетљиве, захтевне и комплексне, неуронске мреже показале су се као елегантно решење механизама обраде фотографије. Уз помоћ малог скупа података обезбеђена је релативно велика прецизност излазних резултата и циљеви рада су постигнути. Показано је да вештачке неуронске мреже могу условно поспешити понашање слично човековом, од којег се захтева да интуитивно, својим осетилима вида, процени да ли је потребно кочити, убрзати или одржавати растојање.

Иако је решење имплементирано у симулационом окружењу КАРЛА симулатора, на основу приказаних резултата из поглавља Резултати, на основу шест имплементираних симулационих случајева, може се доћи до закључка да је овакав један систем могуће реализовати у модерним напредним системима за асистенцију возачу. Без обзира на то, овај рад представља поједностављен модел таквог система, јер у реалном систему, овај систем би морао да узме много више фактора и информација у обзир. На пример, како се систем ослања на РГБ камеру као извор фотографија са пута, морало би се пронаћи решење које би обезбедиле његово несметано функционисање у разним временским

условима и условима смањене видљивости. Важно је поменути да је у овој имплементацији, конволутивна неуронска мрежа пројектована и обучавана од почетка. Генерисан је скуп података од 16568 фотографија које су ручно означене за потребе обучавања неуронске мреже, развијени су сви потребни алати за манипулисање скупом података, али и додатне компоненте потребне за извршавање самог система и симулација.

Придодавање значају развијања оваквих система аутономне вожње увелико би олакшао људима у свакодневним и дужим путовањима где би се човек током пута могао фокусирати на важније ствари од самог управљања возилом или чак одморити до доласка на жељену дестинацију. Поред тога, добра имплементација оваквих система обезбедила би већу безбедност возила, путника и осталих учесника у саобраћају.

8 Референце

[1] L. Xiao and F. Gao, “A comprehensive review of the development of adaptive cruise control systems,” *Vehicle System Dynamics*, vol. 48, no. 10, pp. 1167–1192, Oct. 2010, doi: 10.1080/00423110903365910.

[2] M. W. Libbrecht and W. S. Noble, “Machine learning applications in genetics and genomics,” *Nature Reviews Genetics*, vol. 16, no. 6. Nature Publishing Group, pp. 321–332, May 18, 2015. doi: 10.1038/nrg3920.

[3] H. Ghoddusi, G. G. Creamer, and N. Rafizadeh, “Machine learning in energy economics and finance: A review,” *Energy Economics*, vol. 81, pp. 709–727, Jun. 2019, doi: 10.1016/j.eneco.2019.05.006.

[4] J. T. McCoy and L. Auret, “Machine learning applications in minerals processing: A review,” *Minerals Engineering*, vol. 132. Elsevier Ltd, pp. 95–109, Mar. 01, 2019. doi: 10.1016/j.mineng.2018.12.004.

[5] A. Luckow, M. Cook, N. Ashcraft, E. Weill, E. Djerekarov, and B. Vorster, “Deep learning in the automotive industry: Applications and tools,” in *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*, 2016, pp. 3759–3768. doi: 10.1109/BigData.2016.7841045.

[6] M. Cherian and S. P. Sathiyan, “Neural Network based ACC for Optimized Safety and Comfort,” 2012.

[7] J. David, P. Brom, F. Starý, J. Bradáč, and V. Dynybyl, “Application of artificial neural networks to streamline the process of adaptive cruise control,” *Sustainability (Switzerland)*, vol. 13, no. 8, Apr. 2021, doi: 10.3390/su13084572.

[8] S. S. Mousavi, M. Schukat, and E. Howley, “Deep Reinforcement Learning: An Overview,” *Lecture Notes in Networks and Systems*, vol. 16, pp. 426–440, 2018, doi: 10.1007/978-3-319-56991-8_32.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[10] BOLDENTHUSIAST, “NEURAL NETWORKS AND DEEP LEARNING: AN OVERVIEW.” https://alphabold.com/neural-networks-and-deep-learning-an-overview/

[11] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[12] L. Tóth, “Phone recognition with hierarchical convolutional deep maxout networks,” *Eurasip Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, 2015, doi: 10.1186/s13636-015-0068-3.

[13] I. Wallach, M. Dzamba, and A. Heifets, “AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery,” pp. 1–11, 2015.

[14] M. Z. Alom *et al.*, “A state-of-the-art survey on deep learning theory and architectures,” *Electronics (Switzerland)*, vol. 8, no. 3, pp. 1–67, 2019, doi: 10.3390/electronics8030292.

[15] S. Verma, “Understanding 1D and 3D Convolution Neural Network | Keras.” https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610

[16] D. Nautiyal, “Underfitting and Overfitting in Machine Learning.” https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

[17] ASAM e.V., “ Open Dynamic Road Information for Vehicle Environment ,” *https://www.asam.net/standards/detail/opendrive/*, 2022.

[18] Open-source, “CARLA Simulator,” *https://carla.org/*, 2022.

[19] CARLA, “CARLA 0.9.12 API,” *https://carla.readthedocs.io/en/0.9.12/start_introduction/*, 2022.

[20] CARLA, “CARLA github repository,” *https://github.com/carla-simulator/carla*, 2022.

[21] P. O. U. Sing and M. O. E. Nsembles, “Epo Pt : L Earning R Obust N Eural N Etwork,” vol. 1, no. 2, pp. 1–15, 2017.