

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Simulátor pro průjezd křižovatkou



2021

Vedoucí práce: doc. RNDr. Miroslav Kolařík, Ph.D.

Kristína Poláková

Studijní obor: Informatika, prezenční forma

Bibliografické údaje

Autor: Kristína Poláková
Název práce: Simulátor pro průjezd křižovatkou
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2021
Studijní obor: Informatika, prezenční forma
Vedoucí práce: doc. RNDr. Miroslav Kolařík, Ph.D.
Počet stran: 43
Přílohy: 1 CD
Jazyk práce: slovenský

Bibliographic info

Author: Kristína Poláková
Title: Crossroad Simulator
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2021
Study field: Computer Science, full-time form
Supervisor: doc. RNDr. Miroslav Kolařík, Ph.D.
Page count: 43
Supplements: 1 CD
Thesis language: Slovak

Anotácia

Práca sa zaoberá vytvorením algoritmu na zefektívnenie prejazdu križovatkou riadenou svetelnými signálmi za použitia neurónových sietí. Algoritmus je demonštrovaný na simulátore križovatky. Súčasťou práce je porovnanie efektivity s križovatkou využívajúcou klasický algoritmus, ktorý vie simulátor taktiež simulovať.

Synopsis

The thesis deals with creating effective algorithm for traffic light controlled crossroad using neural networks. This algorithm is demonstrated by crossroad simulator application. The thesis also contains comparism between crossroad using our new algorithm and crossroad using classic algorithm. Both of these algorithms can be simulated by the crossroad simulator.

Kľúčové slová: neurónová sieť; svetelné signály; križovatka; simulátor križovatky

Keywords: neural network; traffic lights; crossroad; crossroad simulator

Ďakujem doc. RNDr. Miroslavovi Kolaříkovi, Ph.D. za vedenie a pomoc pri tvorbe
mojej práce.

*Čestne vyhlasujem, že som celú prácu vrátane príloh vypracovala samostatne a
za použitia iba zdrojov spomínaných v texte práce a uvedených v zozname litera-
túry.*

dátum odovzdania práce

podpis autora

Obsah

1	Úvod	8
2	Križovatka riadená svetelnými signálmi	9
2.1	Signálne systavy	9
2.1.1	Trojfarebná sústava signálov	9
2.1.2	Dvojfarebná sústava signálov	9
2.1.3	Ďalšie signály	10
2.2	Signálny plán	10
2.2.1	Fázy	10
2.2.2	Medzičas	11
2.2.3	Dĺžka cyklu	12
3	Neurónová sieť	13
3.1	Neurón	13
3.1.1	Vstup a výstup neurónu	14
3.2	Predpovedanie neurónovej siete	15
3.3	Učenie neurónovej siete	16
3.3.1	Inicializácia siete	16
3.3.2	Cvičné dáta	17
3.3.3	Predpovedanie	17
3.3.4	Výpočet chyby	17
3.3.5	Spätná propagácia	18
4	Tvorba algoritmu	21
4.1	Signálny plán	22
4.2	Neurónové siete	24
4.2.1	Sieť predvídajúca podľa stavu križovatky	24
4.2.2	Sieť predvídajúca podľa hustoty premávky	25
4.2.3	Kontrola	27
4.3	Križovatka s pevným režimom	27
5	Simulátor križovatky	29
5.1	Funkčnosť aplikácie	29
5.2	Časti aplikácie	29
6	Porovnanie	34
	Záver	35
	Conclusions	36
A	Prehľad simulácií	37
B	Obsah priloženého CD	42

Zoznam obrázkov

1	Ukážka zistenia vzdialenosti kolíznej plochy [3]	11
2	Spojenie dvoch neurónov	13
3	Príklad neurónovej siete	14
4	Funkcia sigmoid	14
5	Výpočet vstupu a výstupu neurónu [1]	15
6	Hľadanie ideálnej váhy pomocou gradientného zostupu	19
7	Využitá križovatka	21
8	Výber typu simulácie	30
9	Nastavenie doby trvania simulácie	30
10	Nastavenie počiatočného počtu áut v pruhoch	31
11	Hlavné okno	31
12	Spôsobenie nehody	32
13	Aktivácia prechodu	32
14	Prejazd vlaku	33
15	Informácie o behu simulácie	33
16	Simulácia 1	37
17	Simulácia 2	37
18	Simulácia 3	38
19	Simulácia 4	38
20	Simulácia 5	39
21	Simulácia 6	39
22	Simulácia 7	40
23	Simulácia 8	40
24	Simulácia 9	41
25	Simulácia 10	41

Zoznam tabuliek

1	Hodnoty na výpočet medzičasu [3]	11
2	Príklad vstupu a výstupu prvej neurónovej siete	25
3	Príklad vstupu a výstupu druhej neurónovej siete	26
4	Porovnanie	34

1 Úvod

Križovatky riadené svetelnými signálmi majú zvyčajne presne určené, kedy má ktorý jazdný pruh zelenú a ako dlho bude trvať. Hovoríme o takzvanom pevnom signálnom pláne. Pri nahromadení áut v cestných pruhoch počas dopravnej špičky sa môžu tvoriť kolóny, čím dôjde k spomaleniu premávky. Riešením tohto problému môže byť využitie voľne tvoreného signálneho plánu, kde je poradie fáz a trvanie zeleného signálu určené krátkodobo podľa účastníkov premávky.

Táto práca sa zaoberá vytvorením algoritmu, ktorý pomocou neurónových sietí dokáže nájsť čo najlepšie prepínanie semaforov a trvanie zelenej fázy tak, aby doprava cez trojprúdovú križovatku plynula čo najľahšie. Do úvahy berieme celkovú situáciu na križovatke – nehody, prejazd vlaku, prechod chodcov a hustotu premávky v jednotlivých pruhoch.

V rámci práce je vytvorená desktopová aplikácia simulujúca prejazd trojprúdovou križovatkou v čase dopravnej špičky, ktorá na prepínanie semaforov využíva náš nový algoritmus, no vie použiť aj klasicý režim prepínania. Následne sú tieto dva algoritmy porovnané.

2 Križovatka riadená svetelnými signálmi

Križovatka riadená svetelnými signálmi využíva na riadenie premávky svetelné signály (semafore). Klasický semafor pre vozidlá je trojfarebný a pre chodcov dvojfarebný. Semafore sa nachádzajú nad vozovkou alebo pri nej, aby boli viditeľné pre účastníkov premávky, ktorým sú určené.

Okrem tohto typu existuje križovatka riadená dopravnými značkami a križovatka bez značiek aj bez svetelných signálov, kde sa účastníci premávky riadia pravidlom pravej ruky.

2.1 Signálne sústavy

Pozrieme sa na typy signálnych sústav [2], ktoré sa na našej križovatke vyskytujú.

2.1.1 Trojfarebná sústava signálov

Signály na riadenie premávky vozidiel (osobné autá, nákladné autá, atď.).

1. Signál *Stoj!* – červené svetlo na semafore. Dáva povinnosť zastaviť vozidlo pred semaforom v dostatočnej vzdialenosti.
2. Signál *Pozor!* – dve varianty:
 - Oranžové, resp. žlté svetlo spolu s červeným svetlom – dáva povinnosť pripraviť sa k jazde.
 - Oranžové, resp. žlté svetlo – dáva povinnosť zastaviť pred semaforom v dostatočnej vzdialenosti, ak je však už veľmi blízko semaforu, môže pokračovať v jazde kvôli bezpečnosti.

Z dôvodu väčšej prehľadnosti grafického znázornenia v simulátore používame na oba tieto signály rovnaké oranžové svetlo.

3. Signál *Volno.* – zelené svetlo na semafore. Dáva možnosť pokračovať v jazde v smere, ktorý je pre príslušný pruh určený.

2.1.2 Dvojfarebná sústava signálov

Signály na riadenie premávky chodcov.

1. Signál *Stoj!* – červené svetlo na semafore. Dáva chodcovi povinnosť nevstupovať do vozovky. Ak sa ešte chodec nachádza na vozovke už počas tohto signálu, môže prechádzanie vozovky dokončiť.
2. Signál *Volno.* – zelené svetlo na semafore. Dáva chodcovi možnosť vstúpiť do vozovky a prejsť ju.

2.1.3 Ďalšie signály

Signály využité pri riadení premávky dopravných prostriedkov s prednosťou jazdy, hlavne vozidiel vychádzajúcich z miesta mimo vozovky, v našom prípade vlakov.

1. Signál *dvoch prerušovaných červených svetiel vedľa seba* - dáva povinnosť zastaviť vozidlo pred semaforom v dostatočnej vzdialenosti.

2.2 Signálny plán

Signálny plán je algoritmus prepínania svetelných zariadení [2]. Určuje poradie a dĺžku trvania signálu voľno jednotlivých signalizačných skupín.

Signalizačná skupina je súbor pohybov, ktoré nie sú navzájom kolízne, teda môžu mať signál voľno naraz.

Pri tvorbe signálneho plánu musíme určiť fázy, dĺžku trvania cyklu, dĺžku trvania signálu voľno a vypočítať medzičasy.

Druhy signálneho plánu:

1. Pevný signálny plán

Fázy, ich poradie aj doba trvania signálu sú pevne určené, nemenia sa krátkodobou. Zmenu poradia fáz môže zapríčiniť udalosť, o ktorej vopred vieme, že môže nastať - napríklad prejazd vlaku.

2. Dynamický signálny plán

Fázy, ich poradie a doba trvania signálu voľno sú určené krátkodobou podľa účastníkov premávky - napríklad hustota premávky v jednotlivých pruhoch.

2.2.1 Fázy

Fáza je časový úsek, v ktorom majú voľno niektoré vzájomne nekolízne dopravné pohyby [2]. Ich tvorením sa snažíme predísť zrážkam pri prejazde vozidiel a prechode chodcov cez križovatku. Pri návrhu poradia fáz sa snažíme o čo najefektívnejšie poradie s ohľadom na intenzitu jednotlivých druhov dopravy, rozloženiu križovatky atď.

Počet fáz je daný rozdelením dopravných pohybov na križovatke, pričom sa snažíme o čo najjednoduchšie rozdelenie.

Poradie fáz je určené viacerými faktormi, uvedieme si tie, ktoré sa vzťahujú na našu križovatku:

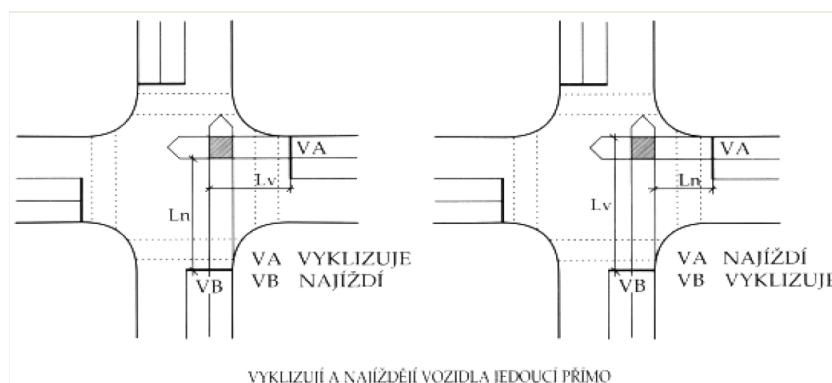
- Nasledovanie niektorých fáz po sebe kvôli plynulosti signálu voľno

- Nasledovanie niektorých smerov po sebe, aby nedošlo k nahromadeniu vozidiel

Fázový prechod je časový interval medzi signálmi voľno skupiny končiacej fázu a skupiny začínajúcej fázu. Môžu byť pevné (pevný SP) alebo pružné (dynamický SP).

2.2.2 Medzičas

Medzičas t_m je časový úsek medzi začiatkom, resp. koncom, signálov *voľno* kolíznych skupín [2]. V tomto intervale musí posledné vozidlo práve signalizujúcej skupiny prejsť kolíznou plochou, kým do nej vstúpi vozidlo z nasledujúcej kolíznej skupiny. Pri výpočte medzičasu sú využívané hodnoty z tabuľky (viz tab. 1). Každá signálna skupina má medzičas určený ako maximálny spomedzi medzičasov vypočítaných pre jej pohyby.



Obr. 1: Ukážka zistenia vzdialenosti kolíznej plochy [3]

VÝJAZDOVÁ A NÁJAZDOVÁ RÝCHLOSŤ	m/s
motorové vozidlá - priamy smer	9.7
motorové vozidlá - oblúk	7.0
chodci	1.4
DĚLKA VOZIDLA	m
motorové vozidlá	5
chodci	0
BEZPEČNOSTNÁ DOBA	s
motorové vozidlá	2
chodci	0

Tabuľka 1: Hodnoty na výpočet medzičasu [3]

Uvedieme si vzorce potrebné na výpočet medzičasu t_m [3]:

$$t_m = t_v - t_n + t_b$$

t_v doba výjazdu
 t_n doba nájazdu
 t_b bezpečnostná doba (zistíme z tab. 1)

$$t_v = \frac{L_v + l_{voz}}{V_v}$$

L_v dĺžka výjazdovej dráhy
 l_{voz} dĺžka vozidla
 V_v rýchlosť výjazdu

$$t_n = \frac{L_n}{V_n}$$

L_n dĺžka odjazdovej dráhy
 V_n rýchlosť nájazdu

2.2.3 Dĺžka cyklu

Cyklus [2] je interval, v ktorom sa prestriedajú všetky fázy, teda úsek od začatia prvej fázy po koniec poslednej fázy. Čas trvania cyklu je vypočítaný ako súčet potrebných dôb trvania signálov voľno a medzičasov príslušných k jednotlivým signálom voľno.

Minimálna dĺžka je 30 sekúnd a optimálna je 50 - 80 sekúnd.

Okrajové hodnoty dĺžky signálu voľno a pozor:

- signál voľno - min. 5 sekúnd
- signál pozor - min. 3 sekundy

3 Neurónová sieť

Neurónová sieť, resp. umelá neurónová sieť, je výpočtový model simulujúci spracovanie informácií prirodzenou neurónovou sieťou, teda ľudským mozgom [1]. Z hľadiska informatiky patrí pod technológie umelej inteligencie a strojového učenia.

Mozog si vie poradiť s úlohami, s ktorými by si bežný neinteligentný model neporadil a neurónové siete nám preto ponúkajú veľa možností na využitie, napríklad pri rozpoznávaní tvarov na obrázku alebo pri klasifikácii.

Základom neurónových sietí je možnosť komunikácie medzi neurónmi a možnosť učiť sa na známych dátach. Na základe naučených vzorcov myslenia sa snaží predpovedať výstupy u neznámych vstupných informácií.

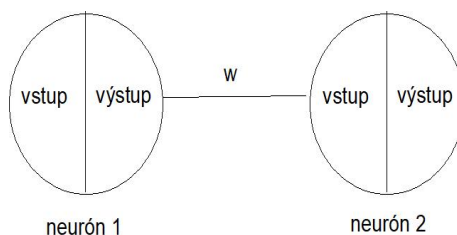
Existuje veľa druhov neurónových sietí, my sa budeme zaoberať najzákladnejším typom a to *doprednými* neurónovými sieťami. Ich hlavnou črtou je, že dáta sa v nich spracovávajú v jednom smere, teda nie sú v nej cyklické spojenia neurónov.

3.1 Neurón

Neurón je základným prvkom neurónovej siete [1]. Je to štruktúra, ktorá má svoj vstup, výstup a vie, s ktorými neurónmi je spojená.

Prirodzený neurón prenáša dáta pomocou synáps, čo sú spojivá medzi neurónmi, a samotnú komunikáciu neurónov zabezpečujú chemické látky a elektrické impulzy.

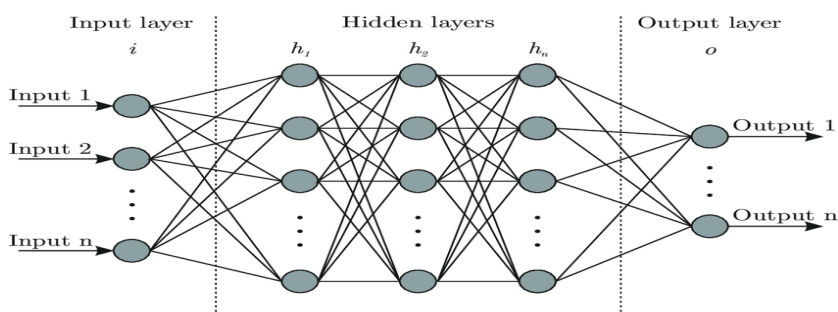
Umelý neurón musí byť navrhnutý tak, aby dokázal všetko, čo prirodzený. Spojenie medzi neurónmi je zvyčajne reprezentované pomocou hodnôt, ktoré nazývame váhy. Tieto hodnoty predstavujú silu spojenia, teda nám nahrádzajú kvantitu chemických látok a silu impulzu. Násobíme nimi výstupy neurónov na ceste k ďalším neurónom. Každé dva neuróny majú vlastnú váhu spojenia (viz obr. 2). Budeme ich značiť w (z angl. weights).



Obr. 2: Spojenie dvoch neurónov

Neuróny sa zoskupujú do vrstiev a tieto sú navzájom poprepájané. Vrstvy majú určené poradie, tak ako aj neuróny v nich, a preto môžeme povedať, že všetky neuróny vo vrstve i sú spojené so všetkými neurónmi vo vrstve $i + 1$. Každá sieť má minimálne dve vrstvy: *vstupnú* (prvú) a *výstupnú* (poslednú). Vrstvy medzi nimi sú *skryté*. Počet vrstiev a počet neurónov v nich určíme

na základe problému, ktorý bude sieť riešiť. Budeme používať viacvrstvové siete (viz obr. 3).

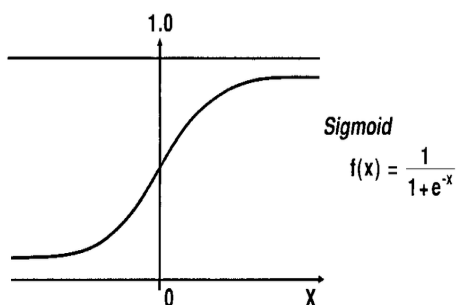


Obr. 3: Príklad neurónovej siete

zdroj: https://www.researchgate.net/figure/Artificial-neural-network-architecture-ANN-i-h-1-h-2-h-n-o_fig1_321259051

3.1.1 Vstup a výstup neurónu

Každá vrstva má svoju aktivačnú funkciu [1]. *Aktivačná funkcia* je aplikovaná na vstup neurónu, aby sme získali jeho výstup. Zabezpečuje nám, že hodnoty posielané medzi neurónmi sa pohybujú iba v intervale funkcie, čo zlepšuje spoľahlivosť siete. Každá vrstva môže mať inú aktivačnú funkciu, najznámejšie sú *sigmoid* a *cross entropy*. My budeme využívať sigmoid ¹ (viz obr. 4).



Obr. 4: Funkcia sigmoid

zdroj: https://www.researchgate.net/figure/An-illustration-of-the-signal-processing-in-a-sigmoid-function_fig2_239269767

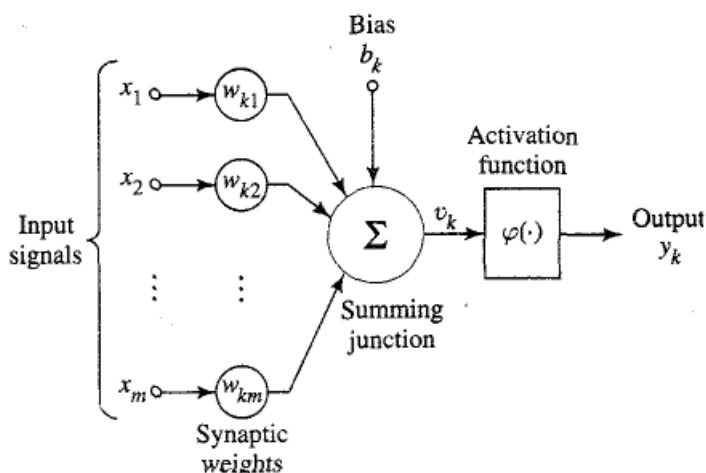
Neurón môže mať takzvaný *bias*, čo je hodnota, ktorá je nezávislá na predchádzajúcej vrstve. Pomáha stabilizovať aktivačnú funkciu tým, že ho pred jej aplikáciou pripočítame ku vstupu. Bias môže byť reprezentovaný ako špeciálny neurón so vstupom 1, ktorý nie je spojený s predchádzajúcou, a ktorého váhy spojení s ďalšou vrstvou predstavujú hodnoty biasu príslušných neurónov [1].

¹viac informácií o funkcii sigmoid dostupné z: <https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function>

Vstupom neurónu vo vstupnej vrstve je hodnota, ktorú mu predáme. Táto hodnota by mala byť normalizovaná tak, aby s ňou neurónová sieť vedela pracovať. Najvhodnejšie je, aby bola hodnota v intervale používanej aktivačnej funkcie.

Vstup neurónu v skrytých vrstvách a v poslednej je súčet hodnôt výstupov neurónov predchádzajúcej vrstvy vynásobených váhami príslušných spojení.

Výstupom neurónu je hodnota aktivačnej funkcie aplikovanej na vstup a prípadný bias. Vo vstupnej vrstve môžeme ako výstup neurónu vydať jej vstup, ak je vhodne normalizovaný, ako je spomenuté vyššie.



Obr. 5: Výpočet vstupu a výstupu neurónu [1]

3.2 Predpovedanie neurónovej siete

O neurónovej sieti hovoríme, že svoj výstup predpovedá, pretože ho hľadá na základe naučených informácií a môže sa zmýliť. Proces predpovedanie môže byť rôzny, podľa typu siete. Ako sme už zmienili, zaoberáme sa doprednými sieťami, u ktorých predpovedanie nazývame aj *feedforwarding* (z angl. feedforward neural networks), teda dopredné necyklické predávanie dát.

Vstupom neurónovej siete sú hodnoty vstupov neurónov vo vstupnej vrstve. Tieto hodnoty predávame sieti zoradené tak, aby sa dostali k správne neurónu.

Výstupom neurónovej siete sú výstupy neurónov výstupnej vrstvy. Tieto hodnoty sú zoradené podľa neurónov, ktoré ich vydali.

Proces predpovedania

1. Predanie vstupných dát vstupnej vrstve.

2. Spracovanie dát neurónmi vstupnej vrstvy a poslanie ich výstupov do prvej skrytej vrstvy.
3. Spracovanie dát neurónmi skrytej vrstvy a poslanie výstupov ďalšej vrstve. Opakuje sa až kým sa nedostaneme do výstupnej vrstvy.
4. Neuróny výstupnej vrstvy spracujú dáta a dajú nám výstupy, ktoré dokopy tvoria výstup našej siete.

3.3 Učenie neurónovej siete

Neurónová sieť simuluje chovanie nášho myslenia a teda nie je úplne presná. Aby predpovedala výstupy čo najlepšie, musíme ju to naučiť. Na tento účel používame dáta, o ktorých vieme, že sú určite pravdivé. Sieť si podľa nich nájde vzor, ktorý bude používať pri predpovedaní výstupov u neznámych vstupných dát.

Postup učenia:

1. Inicializácia siete
2. Tvorba cvičných dát
3. Prepovedanie
4. Výpočet chyby
5. Spätná propagácia

Kroky 3 - 5 sa opakujú kým celková chyba siete nie rovná alebo menšia než nejaká zvolená hodnota alebo sa opakujú určitý počet krát. Opakovania nazývame epochy.

Budeme využívať obdobu stochastickej metódy učenia [5]. U tejto metódy sa v každej epoche nepracuje s celou množinou cvičných dát, ale náhodne sa vyberie podmnožina, podľa ktorej sa upravujú váhy. Táto metóda by mala byť pri učení neurónovej siete efektívnejšia. My budeme v každej epoche pracovať so všetkými cvičnými dátami, avšak chybu budeme počítat pre každú položku zvlášť.

3.3.1 Inicializácia siete

Pri inicializácii siete jej dávame informácie o vrstvách a inicializujeme váhy.

Počet vrstiev a počet ich neurónov je pri učení a samotnom fungovaní siete veľmi dôležitým faktorom. Mali by sme sa zamyslieť, čo vlastne naša sieť robí a ako si tento proces môžeme predstaviť. U niektorých typov problémov to však nie je jednoduché a preto sa nám zvyčajne nepodarí nájsť optimálny počet na prvý krát. Pomôcť nám môže napríklad fakt, že počet neurónov je v skrytých vrstvách bežne väčší než vo vstupnej a výstupnej vrstve.

Zdalo by sa, že vyšším počtom vrstiev a neurónov sa zvýši presnosť siete, no toto tvrdenie všeobecne neplatí.

Je dobré zamyslieť sa, či problém, ktorý má sieť riešiť, nevieme rozdeliť na niekoľko menších častí, aby sme na riešenie mohli využiť viacero sietí. Siete sa ľahšie naučia riešiť jednoduchšie problémy.

Hodnotu váh pri inicializácii vyberáme náhodne z nejakého intervalu. Váhy sú pri predpovedaní výstupu neurónovej siete najdôležitejšie a práve tieto hodnoty upravujeme pri učení.

3.3.2 Cvičné dáta

Vhodné zvolenie cvičných dát je pri učení neurónovej siete kľúčové. Dáta využívané na učenie neurónovej siete pozostávajú zo známych vstupov a im príslušných výstupov, o ktorých vieme, že sú korektné.

Je dôležité, aby cvičné dáta splňali nasledujúce:

- Sú realistické. Každý vstup by mal byť príkladom dát, ktoré môže pri predpovedaní naozaj dostať a každý výstup musí byť príkladom dát, ktoré neurónová sieť naozaj môže vydať. Taktiež musíme prihliadať na počet neurónov vo vstupnej a výstupnej vrstve.
- Je ich dostatočný počet. Musíme sa zamyslieť, koľko možných vstupov môže sieť dostať. Čím viac možností je, tým viac cvičných dát by sme mali sieti poskytnúť. Obecne jej nedávame všetky možné vstupy, aby sme zistili, či sieť po naučení naozaj dokáže predpovedať výstupy aj pre ešte neznáme dáta.
- Vhodné usporiadanie dát vstupu. Keďže sa sieť učí na základe rozpoznávania vzorov v dátach, mali by sme si premyslieť, ako ich usporiadať, aby neboli chaotické.

3.3.3 Predpovedanie

Predpovedanie výstupu na základe vstupu je popísané v časti 3.2. Vezmeme jeden zo vstupov cvičných dát a neurónová sieť ho postupne spracuje až sa dostaneme do výstupnej vrstvy. Jej výstupy porovnáme s požadovaným výstupom prislúchajúcim ku vstupu a vypočítame chybu, ktorú budeme spätne propagovať.

3.3.4 Výpočet chyby

Chybu predpovede neurónovej siete môžeme počítať viacerými spôsobmi. Existujú napríklad: *mean squared error* (stredná kvadratická chyba), *cross entropy* (krížová entropia) alebo *quadratic error* (kvadratická chyba). Každá z nich je

využitelná na iný problém ². My budeme používať *mean squared error*, ktorú budeme značiť E .

Chyba E sa zvyčajne používa u sietí predpovedajúcich len jednu hodnotu, teda s jedným výstupným neurónom. My budeme mať vo výstunej vrstve viac neurónov, takže budeme chybu počítat pre každý neurón zvlášť.

Vzorec výpočtu chyby je

$$\forall j : E = \frac{1}{n} \sum_{i=1}^n (\text{idealOut}_{ij} - \text{realOut}_{ij})^2$$

kde ij je j -tý prvok i -tej dvojice cvičných dát, idealOut je požadovaný výstup neurónovej siete, realOut je reálny výstup neurónovej siete, n je počet cvičných dát, resp. počet dát, ktorý na výpočet používame.

3.3.5 Spätaná propagácia

Spätaná propagácia (angl. backpropagation) je proces, pri ktorom chybu vypočítanú v poslednej vrstve predávame spätne cez všetky skryté vrstvy až po vstupnú [4][6].

Každý neurón vo vrstve i je spojený s každým neurónom vo vrstve $i - 1$ a všetky spojenia majú svoju váhu, ktorá sa zúčastnila na vytvorení chyby vo výstupnej vrstve. Tieto váhy teda musíme upraviť s prihliadnutím na chyby vo všetkých neurónoch vo vrstve i , čím sa snažíme doceliť zminimalizovanie E . Budeme využívať metódu stochastického gradientného zostupu (angl. gradient descent).

Pri metóde gradientného zostupu hľadáme gradient chyby E , ktorá je vlastne funkciou (viz obr. 6). Gradient je vektor parciálnych derivácií chyby podľa jednotlivých váh. Gradient nám ukáže smer, v ktorom chyba rastie. Hodnotu váh musíme upraviť tak, aby sme sa pohli v opačnom smere. U tejto metódy vlastne hľadáme minimum funkcie chyby E .

Výstupná vrstva

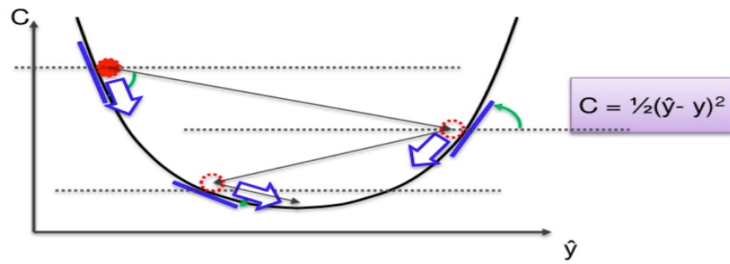
Gradient ∇E každého neurónu vypočítame nasledujúcim vzorcom, kde w_1, \dots, w_n sú váhy, spájajúce výstupnú vrstvu s predchádzajúcou, a n je počet dvojíc vstupov a výstupov v cvičných dátach alebo počet častí cvičných dát, ak používame stochastickú metódu:

$$\nabla E = \left(\frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n} \right)$$

Túto hodnotu vieme vypočítat využitím reťazového pravidla:

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial \text{out}_i} \cdot \frac{\partial \text{out}_i}{\partial \text{in}_i} \cdot \frac{\partial \text{in}_i}{\partial w_i}$$

²viac informácií o chybách dostupné z: <https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/>



Obr. 6: Hľadanie ideálnej váhy pomocou gradientného zostupu
 zdroj: <https://www.superdatascience.com/blogs/artificial-neural-networks-stochastic-gradient-descent/>

Hodnota hovoriaca o tom, ako sa zmení E , ak sa zmení výstup neurónu, pričom prihliadame na to, že pracujeme len s jednou položkou cvičných výstupov, čo nám výpočet zjednoduší:

$$\begin{aligned} \frac{\partial E}{\partial out_i} &= \frac{\partial}{\partial out_i} E = \frac{\partial}{\partial out_i} (idealOut_i - out_i)^2 \\ &= 2(out_i - idealOut_i) \end{aligned}$$

Hodnota hovoriaca o tom, ako sa zmení výstup, ak sa zmení vstup, keď vy-užívame funkciu sigmoid:

$$\frac{\partial out_i}{\partial in_i} = \frac{\partial}{\partial in_i} \cdot \frac{1}{1 + e^{-(in_i)}} = \frac{1}{1 + e^{-(in_i)}} \cdot \left(1 - \frac{1}{1 + e^{-(in_i)}}\right)$$

Hodnota hovoriaca o tom, ako sa zmení vstup, ak sa zmení váha, kde jp znamená j -tý neurón p -tá (predošlá) vrstva:

$$\frac{\partial in_i}{\partial w_i} = \frac{\partial}{\partial w_i} \sum_j^n w_j out_{jp} = out_{ip}$$

Na výpočet novej váhy potrebujeme poznať rozdiel medzi novou a starou váhou, kde γ je rýchlosť učenia, ktorú si zvolíme pred začatím učenia:

$$\Delta w_i = -\gamma \frac{\partial E}{\partial w_i}$$

Nakoniec upravíme všetky váhy:

$$\forall i : w_i = w_i - \Delta w_i$$

Ešte si zavedieme nasledujúci vzorec, ktorý nám zjednoduší prácu s gradientom v skrytých vrstvách:

$$\delta_i = \frac{\partial E}{\partial out_i} \cdot \frac{\partial out_i}{\partial in_i}$$

Gradient potom môžeme napísať ako:

$$\frac{\partial E}{\partial w_i} = \delta_i \cdot out_{ip}$$

Skryté vrstvy

Vzorec na vypočítanie gradientu v skrytej vrstve je rovnaký ako vo výstupnej.

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial out_i} \cdot \frac{\partial out_i}{\partial in_i} \cdot \frac{\partial in_i}{\partial w_i}$$

Rozdiel je však vo výpočte $\frac{\partial E}{\partial out_i}$, pretože musíme prihliadať aj na vrstvy medzi skrytou a výstupnou vrstvou. V nasledujúcich vzorcoch E_{rj} znamená výpočet gradientu vo vrstve j a neurón r , pričom k je počet neurónov vo vrstve j medzi vrstvami:

$$\frac{\partial E}{\partial out_i} = \sum_{r=1}^k \frac{\partial E_{rj}}{\partial out_i}$$

$$\frac{\partial E_{rj}}{\partial out_i} = \frac{\partial E_{rj}}{\partial in_{rj}} \cdot \frac{\partial in_{rj}}{\partial out_i}$$

Hodnota hovoriaca o tom, ako sa zmení vstup neurónu r vo vrstve j , ak sa zmení výstup neurónu i v skrytej vrstve, pričom n je počet neurónov skrytej vrstvy:

$$\frac{\partial in_{rj}}{\partial out_i} = \frac{\partial}{\partial out_i} \sum_{r=1}^n (out_r \cdot w_r) = w_i$$

Hodnota hovoriaca o tom, ako sa zmení E , ak sa zmení vstup neurónu j :

$$\frac{\partial E_{rj}}{\partial in_{rj}} = \text{už vypočítané vo vrstve } j \text{ ako } \delta_i, \text{ preznačíme ho na } \delta_{rj}$$

Gradient teraz môžeme napísať následne, pričom k je počet neurónov v j -tej vrstve:

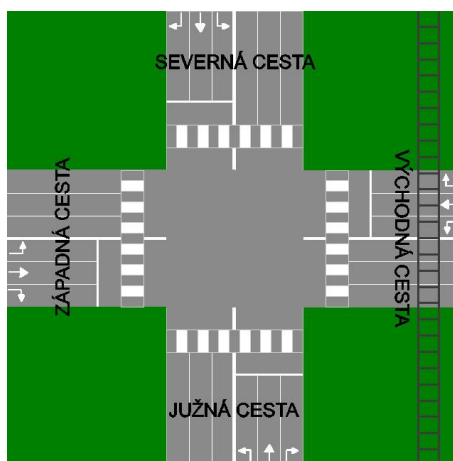
$$\frac{\partial E}{\partial w_i} = w_i \sum_{r=1}^k \delta_{jr}$$

Úprava váh je rovnaká ako vo výstupnej vrstve.

4 Tvorba algoritmu

Bol vytvorený algoritmus, ktorý pre svetelnú križovatku vytvára dynamický signálny plán pomocou neurónových sietí tak, aby bol čo najefektívnejší počas dopravnej špičky. Križovatku využívajúcu tento algoritmus budeme volať *neurónová*.

Zvolili sme si trojprúdovú križovatku (viz obr. 7). Na využívanie neurónových sietí je najlepšia, pretože sa dá dobre využiť hustota premávky v jednotlivých pruhoch, pretože sa autá radia samostatne podľa smeru odbočenia. Údaj o hustote premávky, teda približnom počte áut, by sme pri reálnej križovatke zistili buď použitím senzorov alebo by sme využili neurónovú sieť, ktorá by túto hodnotu vedela podľa viedozáznamu zistiť. Ak by sme použili iný typ križovatky s tým, že by boli niektoré smery radené spolu – najčastejšie to bývajú smer vpravo a rovno – museli by sme tento fakt brať do úvahy, pretože by v týchto pruhoch bolo stále viac áut než v ostatných.



Obr. 7: Využitá križovatka

Križovatka má nasledovné obmedzenia:

1. **Aktívne môžu byť maximálne dva prechody naraz** – pri aktivácii viac ako dvoch prechodov by prišlo k situácii, že by nemohli byť zelené žiadne dopravné smery.
2. **Nehoda môže byť na maximálne jednej ceste** – je možné mať nehodu na každej ceste v mieste pred semaforom, kde k nej môže prísť najčastejšie z dôvodu preradovania sa do iného smeru. Toto obmedzenie sme zvolili preto, že stroj žiadnu nehodu nedokáže vyriešiť, no môže sa pokúsiť zabezpečiť prejazdnosť a istú bezpečnosť. Ak by bola nehoda v inom mieste, než v pruhoch pred semaforom, neurónová sieť má menšiu šancu zaručiť bezpečnosť účastníkov premávky, no môže napríklad automaticky zavolať pomoc, či úplne vypnúť semafory.

3. **Vlak je iba na jednej ceste** – vychádzame zo zložitosti križovatky, ktorú sme si zvolili. Toto obmedzenie je bežné.

4.1 Signálny plán

Fázy našej križovatky určujú neurónové siete.

Ešte pred vytvorením neurónových sietí sme si museli určiť kolízne dopravné pohyby. Miesto vytvorenia kolíznych skupín sa nám však hodilo reprezentovať kolízne smery podľa jednotlivých stavov, resp. zelených pruhov. Reprezentácia je vhodnejšia z hľadiska tvorby cvičných dát pre neurónovú sieť aj kvôli samotným vstupom a výstupom, ktoré nám siete dávajú.

Pokiaľ na ceste niekto aktivuje prechod alebo bude prechádzať vlak, naša sieť musí zastaviť všetky pohyby, ktoré vedú do príslušných ciest a všetky pohyby smerujúce z nich von, aby nedošlo k zrážke.

Pri nehode je riešenie zložitejšie, ako sme už spomenuli vyššie pri obmedzeniach križovatky. Naša sieť sa snaží zabrániť nakopeniu áut v mieste nehody tak, že dá signál *voľno* všetkým pruhom na príslušnej ceste, kde k nehode došlo a ostatným dá červenú. Vozidlá tak budú mať priestor aj v pruhoch protismeru, aby nehodu obišli, kým premávku nepríde niekto riadiť.

Medzičas vypočítame podľa tabuľky a vzorca uvedených v sekcii 2. Budeme počítat iba smery, kde sú kolízne dráhy najdlhšie a vyberieme z nich maximálny medzičas, ktorý využijeme ako medzičas pre celú križovatku. Ak bude táto hodnota nepárne číslo, pridáme k nemu sekundu navyše, aby sa nám s ním dobre pracovalo pri dávaní signálu pozor.

Šírku pruhu si stanovíme na 2.75 m. Výsledné hodnoty medzičasov budeme zaokrúhľovať nahor, pričom hodnoty výsledkov sú v sekundách.

Pohyby vozidiel rovno:

$$\begin{aligned}t_o &= \frac{4 \cdot 2.75 + 5}{9.7} = 1.65 \\t_n &= \frac{3 \cdot 2.75}{9.7} = 0.85 \\t_b &= 2 \\t_m &= 1.65 - 0.85 + 2 = 2.8 \doteq 3\end{aligned}$$

Pohyby vozidla a chodca, kde vozidlo odchádza z cesty a chodec vchádza:

$$\begin{aligned}
t_o &= \frac{7 \cdot 2.75 + 5}{9.7} = 2.5 \\
t_n &= \frac{1 \cdot 2.75}{1.4} = 1.96 \\
t_b &= 1 \\
t_m &= 2.5 - 1.96 + 1 = 1.54 \doteq 2
\end{aligned}$$

Pohyby vozidla a chodca, kde chodec odchádza z cesty a vozidlo vchádza:

$$\begin{aligned}
t_o &= \frac{2 \cdot 2.75 + 0}{1.4} = 3.93 \\
t_n &= \frac{7 \cdot 2.75}{9.7} = 1.7 \\
t_b &= 1 \\
t_m &= 3.93 - 2.14 + 1 = 2.79 \doteq 3
\end{aligned}$$

Pohyby do oblúka:

$$\begin{aligned}
t_o &= \frac{7 \cdot 2.75 + 5}{9.7} = 2.5 \\
t_n &= \frac{0 \cdot 2.75}{7} = 0 \\
t_b &= 2 \\
t_m &= 2.5 - 0 + 2 = 4.5 \doteq 5
\end{aligned}$$

Pohyb do oblúka a rovno:

$$\begin{aligned}
t_o &= \frac{7 \cdot 2.75 + 5}{7} = 3.46 \\
t_n &= \frac{4 \cdot 2.75}{9.7} = 1.14 \\
t_b &= 2 \\
t_m &= 3.46 - 1.14 + 2 = 4.32 \doteq 5
\end{aligned}$$

Maximálny medzičas, teda čas medzi signálmi voľno kolíznych smerov je 5 s, po pridaní jednej sekundy dostaneme medzičas našej križovatky, teda 6 s.

Trvanie cyklu nie je možné u neurónovej križovatky určiť, pretože jej fázy, ich poradie aj dĺžku zeleného signálu určujú neurónové siete, takže sa cyklus dynamicky mení.

Neurónová sieť každých 30 sekúnd vyhodnotí situáciu na križovatke a vydá nám pohyby, ktoré budú mať voľno v ďalšej fáze. Môže sa stať, že niektoré pohyby budú mať voľno aj v niekoľkých po sebe nasledujúcich fázach.

Pohyby nemajú stanovené, kedy budú mať signál voľno. Mohlo by sa zdať, že niektoré pruhy sa na rad nedostanú vôbec, preto sme tento prípad skúsili ošetriť. Ak by pohyb nemal voľno 5 fáz za sebou, neurónovej sieti pošleme miesto reálnych informácií o hustote prevádzky v tomto pruhu, maximálny približný počet áut, ktorý sa aktuálne nachádza v niektorom z pruhov, čím by mal byť pohyb zaradený do skupiny s vyššou prioritou. Nie je však zaručené, že dostane voľno hneď v ďalšej fáze s ohľadom na stav križovatky a kolízne smery, takže tento postup opakujeme až kým nedostane voľno. Pohyb aj vďaka tomuto opatreniu nemusí prísť na rad nikdy, čo by bol znak nesprávne naučenej neurónovej siete, resp. zle zvolenými cvičnými dátami.

4.2 Neurónové siete

Neurónové siete nám predpovedajú pohyby, ktoré budú mať v ďalšej fáze križovatky voľno vzhľadom na stav križovatky (aktívny prechod, prechod vlaku, nehoda) a približný počet áut v jednotlivých smeroch. Ako vidíme, tento problém sa dá pekne rozdeliť do dvoch častí a preto sme použili dve neurónové siete.

Spoločné vlastnosti oboch neurónových sietí:

- dopredné
- viacvrstvové
- aktivačná funkcia sigmoid
- chyba mean squared error

4.2.1 Sieť predvídajúca podľa stavu križovatky

Prvá zo sietí nám na základe stavu križovatky vráti všetky možné pruhy, ktoré môžu byť v ďalšej fáze zelené. Stav križovatky určujeme podľa aktívnych prechodov, nehody alebo prejazdu vlaku.

Vstup a výstup tejto siete je kódovaný nulami a jednotkami:

- **1 vo vstupe** znamená, že na križovatke je na danom mieste aktívny prechod, bude prechádzať vlak alebo je tu nehoda
- **0 vo vstupe** znamená, že na križovatke nie je na danom mieste aktívny prechod, nebude prechádzať vlak a nie je tu nehoda
- **1 vo výstupe** znamená, že daný pruh, resp. pohyb, bude mať v ďalšej fáze signál voľno
- **0 vo výstupe** znamená, že daný pruh, resp. pohyb, bude mať v ďalšej fáze signál stoj

Štruktúra vstupu a výstupu (viz tab. 2) bola zvolená z dôvodu efektívnejšieho učenia siete. Preto máme údaje o vlaku uvedené aj na cestách o ktorých vieme, že na nich nikdy nemôže byť.

cesta	popis vstupu	vstup	výstup	popis výstupu
severná cesta	prechod	1	0	pravý pruh
	nehoda	0	0	stredný pruh
	vlak	0	0	ľavý pruh
východná cesta	prechod	0	0	pravý pruh
	nehoda	0	0	stredný pruh
	vlak	1	0	ľavý pruh
južná cesta	prechod	0	0	pravý pruh
	nehoda	0	0	stredný pruh
	vlak	0	1	ľavý pruh
západná cesta	prechod	0	1	pravý pruh
	nehoda	0	0	stredný pruh
	vlak	0	0	ľavý pruh

Tabuľka 2: Príklad vstupu a výstupu prvej neurónovej siete

Počet neurónových vrstiev sme zvolili na 3 a počet neurónov v nich je postupne 12, 24, 12. U tohto problému bolo ťažšie predstaviť si, ako napasovať kroky jeho riešenia do jednotlivých vrstiev.

Pri tvorbe cvičných dát si pri získavaní výstupu inicializujeme výstup na sáme jednotky, potom podľa vstupných dát postupne prechádzame kolízne smery a odstránime ich z priebežného výstupu.

Zdvojnásobenie počtu neurónov v skrytých vrstvách sa po niekoľkých pokusoch osvedčilo ako najlepšie riešenie.

Cvičné dáta sme pri tejto sieti vytvorili ľahko. Vzhľadom na obmedzenia križovatky sú všetky možné cvičné vstupy a výstupy ľahko zistiteľné. Cvičné výstupy sme hľadali pomocou vlastného programu na generáciu cvičných dát.

Neurónová sieť sa naučila podľa týchto cvičných dát určovať pruhy, ktoré môžu mať voľno, s chybou 0.0001.

4.2.2 Sieť predvídajúca podľa hustoty premávky

Druhá sieť nám na základe dát z prvej siete vyberie pohyby, resp. pruhy, ktoré môžu mať v ďalšej fáze voľno. K nim zistíme počet áut, podľa ktorého je pohyb zaradený do jednej zo skupín označených 1, 2, 3, 4, 5. Kritérium zaradenia do skupiny sa mení podľa maximálneho počtu áut v jednom smere pred výpočtom dát pred ďalšiu fázu. Sieť nám na základe týchto údajov vydá pruhy, ktoré by mali mať v ďalšej zelenej fáze voľno tak, aby nedošlo ku kolízii.

Vstup, teda pohyby zaradené do skupín podľa áut, je kódovaný číslami 0, 0.1, 0.2, 0.3, 0.4, 0.5. Ako vidíme, označenie skupín bolo prepočítané tak, aby patrili do intervalu funkcie sigmoid a sieť s nimi dokázala dobre pracovať.

- **0 vo vstupe** znamená, že daný pohyb nemôže mať v ďalšej fáze voľno, pretože to nedovoľuje stav križovatky (určené podľa výstupu prvej siete)
- **0.1, 0.2, 0.3, 0.4, 0.5 vo vstupe** znamená, že podľa počtu áut patrí daný pohyb do jednej z týchto skupín, pričom sieť by mala dať voľno tým s najvyššou hodnotou, no s ohľadom na kolíznosť

cesta	pruh	vstup	výstup
severná	pravý	0.5	1
	stredný	0.4	1
	ľavý	0.1	0
východná	pravý	0.2	1
	stredný	0.1	0
	ľavý	0.3	1
južná	pravý	0.4	1
	stredný	0.1	0
	ľavý	0.5	1
západná	pravý	0.2	0
	stredný	0.4	0
	ľavý	0.5	0

Tabuľka 3: Príklad vstupu a výstupu druhej neurónovej siete

Výstup je kódovaný nulami a jednotkami.

- **1 vo výstupe** znamená, že daný pruh bude mať v nasledujúcej fáze signál voľno
- **0 vo výstupe** znamená, že daný pruh nesmie mať v nasledujúcej fáze signál voľno kvôli kolízii

Počet vrstiev sme zvolili na 4 a počet neurónov v nich je postupne 12, 24, 24, 12. Ani u tohto problému nebolo jednoduché predstaviť si jednotlivé rozdelenie krokov riešenia do vrstiev.

Pri tvorbe cvičných dát pridáme k výstupu tak, že si najprv inicializujeme výsledok, tentokrát na samé nuly. Postupne budeme prechádzať hodnoty pohybov dané vstupom podľa veľkosti zostupne. Ak má pohyb hodnotu rôznu od nuly, teda môže mať voľno, skontrolujeme, či nie je v kolízii s pruhmi, ktoré už vo výstupe majú daný signál voľno, ak nie je, na príslušné miesto zapíšeme vo výstupe 1, ak je kolízny, nerobíme nič. Ak má pohyb hodnotu nula, preskočíme na ďalší

pohyb. Týmto spôsobom by mali mať v ďalšej fáze voľno pruhy s čo najväčším počtom áut, samozrejme vzhľadom ku kolíziám.

Počet vrstiev bol teda zvolený ako najvhodnejší po niekoľkých pokusoch.

Tvorba **cvičných dát** pre túto sieť bola o niečo zložitejšia. Keďže máme 6 hodnôt, ktoré môžeme mať na vstupe o dvanástich hodnotách, počet všetkých variácií je 6^{12} , čo je veľmi veľa možností. V skutočnosti ich však je oveľa menej, pretože vzhľadom na stav križovatky a proces zaraďovania pohybov do skupín podľa áut, nemôžu niektoré možnosti nastať vôbec a v každom vstupe bude aspoň jeden pruh, ktorý je v najvyššej skupine. Na tvorbu dát sme zvolili náhodnú metódu, ktorá tieto faktory berie do úvahy a náhodne nám vygeneruje 1000 vstupov a k nim výstupy.

Pomocou týchto vstupných dát sme dokázali sieť v 2000 epochách naučiť rozpoznávať, ktoré pruhy majú mať voľno podľa hustoty premávky, s chybou 0.0003.

4.2.3 Kontrola

Neurónové siete nám s malou chybou predpovedajú pohyby, ktoré majú mať voľno. Ich spoľahlivosť závisí na cvičných dátach a dĺžke učenia, no aj tá najlepšie naučená sieť sa môže zmýliť, čo by pri riadení križovatky mohlo mať fatálne následky.

V predpovediach siete kontrolujeme, či nemajú voľno kolízne pohyby, a ak majú, tak kolíznym pohybom s menším počtom áut dáme signál stoj. Nekonrolujeme však, či sieť nezabudla dať voľno niektorému nekolíznemu pohybu, pretože kontrola slúži iba na predídenie zrážke.

Počet chýb si zaznamenávame, aby sme mohli siete preučiť v prípade veľkej chybovosti.

4.3 Križovatka s pevným režimom

Na porovnanie efektivity režimu neurónovej križovatky a klasickej križovatky sme si vytvorili aj pevný signálny plán. Je určený rovnakej trojprúdovej križovatke. Jeho tvorba nie je základným zameraním práce, preto si ho iba stručne popíšeme.

Klasická križovatka bude využívať osem fáz, z toho štyri z nich sú špeciálne a križovatka sa do nich prepne vtedy, ak dôjde k prejazdu vlaku alebo k nehode. Prechody pre chodcov sú určené na pevno.

Fázy sú uložené ako vektory kódované nulami a jednotkami, pričom tieto majú rovnaký význam u výstupov neurónových sietí, preto si ich nebudeme uvádzať. Toto bolo zavedené, aby sme mohli medzi fázami prepínať rovnakou metódou ako u neurónovej križovatky.

Medzičas bol zvolený rovnaký ako u neurónovej križovatky, aby sme mohli pri ich porovnávaní čo najlepšie určiť skutočný rozdiel.

Trvanie cyklu si lahko spočítame z počtu fáz (4), medzičasu(6 sekúnd) a dĺžky signálu voľno (30 sekúnd). Do úvahy neberieme pri výpočte špeciálne fázy, ale len obyčajné. Cyklus teda trvá $4 \cdot 36 = 144$ sekúnd, takže asi 2.4 minúty. Tieto hodnoty boli zvolené, aby korešpondovali s neurónovou križovatkou, ktorá hľadá novú fázu každých 30 sekúnd a jej signál pozor trvá 6 sekúnd.

5 Simulátor križovatky

Na vyskúšanie vytvoreného algoritmu bola vytvorená desktopová *WPF* aplikácia (*Windows Presentation Foundation Application*) vo vývojovom prostredí *Microsoft Visual Studio 2019* pomocou *.NET Frameworku*. Na tvorbu bol použitý jazyk *C#*.

Simulátor mal na prácu s neurónovými sieťami pôvodne využívať knižnicu pre *.NET Framework ML.NET*³ (*Machine Learning .NET*). Po hlbšom preskúmaní sme však zistili, že pre naše ciele nie je vhodná, nakoľko je prevažne určená na detekciu tvarov na obrázku a na riešenie ostatných problémov využíva už predom naučené modely. Nebolo možné upravovať počty neurónov v jednotlivých vrstvách ani počty vrstiev, bez čoho sa pri našom algoritme nezaobídeme. Preto bol na prácu s neurónovými sieťami vytvorený vlastný program v jazyku *C#*, ktorý žiadne špeciálne knižnice nevyužíva.

5.1 Funkčnosť aplikácie

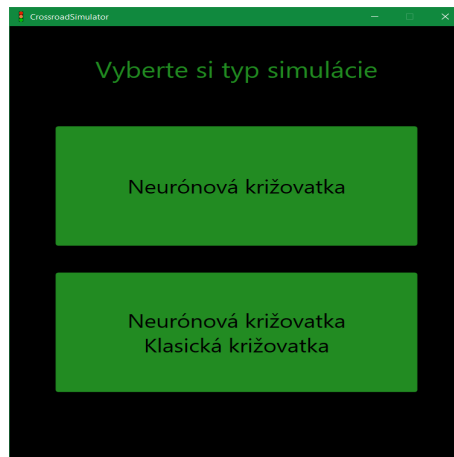
- simulácia neurónovej križovatky
- simulácia klasickej križovatky - kvôli porovnaniu efektivity neurónovej križovatky
- voľba doby trvania simulácie
- nastavenie počiatočného počtu áut v jednotlivých pruhoch
- možnosť pozastaviť, spustiť, zastaviť simuláciu
- interakcia užívateľa s neurónovou križovatkou - aktivácia prechodu, aktivácia prejazdu vlaku a spôsobenie nehody
- interakcia užívateľa s klasickou križovatkou - aktivácia prejazdu vlaku a spôsobenie nehody
- zobrazenie informácie behu simulácie

Jednotlivé funkčnosti si detailnejšie popíšeme v nasledujúcej časti spolu s grafickými ukážkami.

5.2 Časti aplikácie

Hneď po spustení sa zobrazí okno, kde máme na výber z dvoch možností simulácie (viz obr. 8).

³viac informácií o *ML.NET* dostupné z: <https://dotnet.microsoft.com/apps/machinelearning-ai/ml-dotnet>



Obr. 8: Výber typu simulácie

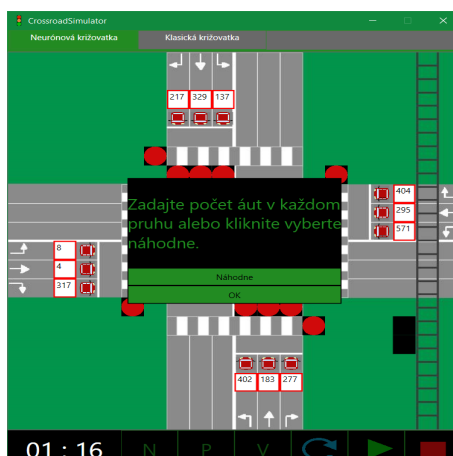
Ak si zvolíme možnosť *Neurónová križovatka*, bude spustená simulácia neurónovej križovatky. Pri výbere *Neurónová križovatka Klasická križovatka* sa spustí simulácia neurónovej aj klasickej križovatky, aby sme mohli ich beh porovnať.

Neurónová križovatka už je naučená, pričom údaje o svojich sieťach si prečíta zo súboru, ktorý vznikol pri jej učení.



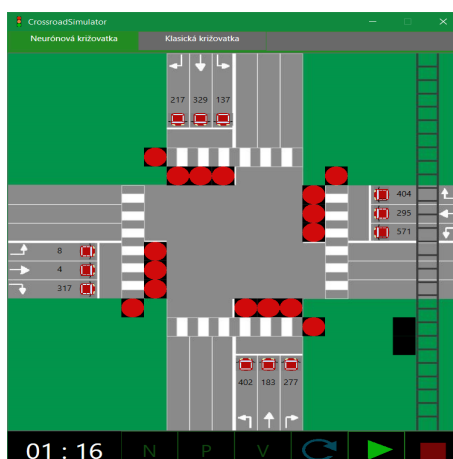
Obr. 9: Nastavenie doby trvania simulácie

Môžeme si nastaviť trvanie simulácie v intervale 1 sekunda až 59 minút 59 sekúnd (viz obr. 9).



Obr. 10: Nastavenie počiatočného počtu áut v pruhoch

Počiatočný počet áut si môžeme zvoliť vlastný alebo si zvolíme náhodný (viz obr. 10). Túto funkciu sme zaviedli preto, aby sme na križovatke nemuseli mať prázdne pruhy a čakať, kým sa dopravná špička vytvorí.



Obr. 11: Hlavné okno

Hlavné okno (viz obr. 11) pozostáva z kontrolného panelu v dolnej časti obrazovky, samotnej plochy na simuláciu a záložiek na prepnutie medzi klasickou a neurónovou križovatkou.

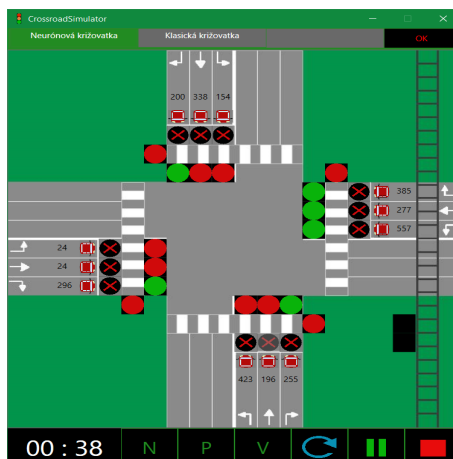
Čísla v pruhoch udávajú počet áut a menia sa podľa prichádzajúcich a odchádzajúcich, pričom autá odchádzajú bežnou rýchlosťou jedno auto za sekundu a prichádzajú náhodne.

Semaforey pre vozidlá sú uložené priamo na ceste kvôli prehľadnosti.

Ovládací panel v dolnej časti okna nám umožňuje simuláciu riadiť. Úplne naľavo máme odpočet času. Na spustenie, pozastavenie a zastavenie sú použité zaužívané znaky. Na spôsobenie nehody, aktiváciu prechodu a prejazd vlaku sú tlačidlá pomenované N, P, V.

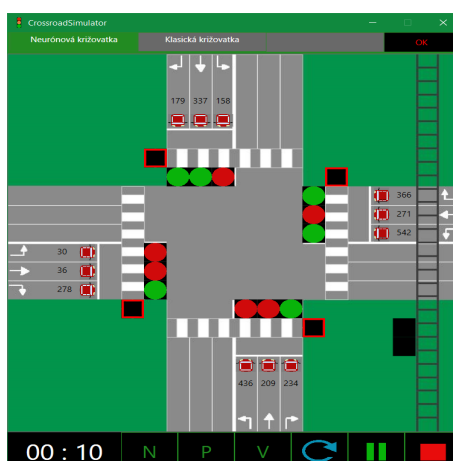
Tlačidlá N, P, V nie sú aktívne počas prepínania fáz, aby sme zabezpečili konzistentnosť dát. Nekonzistentnosť by mohla spôsobiť, že signál voľno by mali kolízne pohyby.

Aktivácia prechodu, spôsobenie nehody a prejazd vlaku nie sú automatické, aby sme si mohli simuláciu prispôbiť našim potrebám, napríklad z dôvodu porovnania. Jediná výnimka je pri klasickej križovatke, na ktorej zapnutie prechodov vyplýva z pevne daných fáz a preto sa nedá nastaviť.



Obr. 12: Spôsobenie nehody

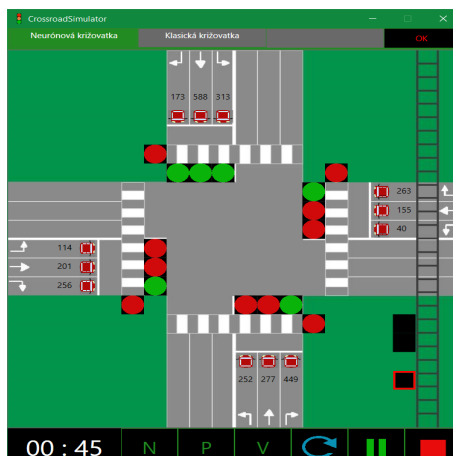
Po kliknutí na tlačidlo N (spôsobiť nehodu): Na každom jazdnom pruhu sa zobrazí červený kríž a tlačidlo OK na potvrdenie voľby v pravej hornej časti križovatky (viz obr. 12). Križovatka nám dovolí vybrať si spôsobenie nehody iba na jednom jazdnom pruhu. Ak si rozmyslíme spôsobenie nehody, stačí opäť kliknúť na tlačidlo N.



Obr. 13: Aktivácia prechodu

Po kliknutí na tlačidlo P (aktivovať prechod): Pri každom semafore jednot-

livých prechodov sa nám zobrazí tlačidlo na jeho aktiváciu na každom semafore pre chodcov a tlačidlo OK na potvrdenie voľby v pravej hornej časti okna (viz obr. 13). Križovatka nám dovolí aktivovať iba dva ľubovoľné semafore naraz. Ak si aktiváciu prechodu rozmyslíme, opäť stisneme tlačidlo P.



Obr. 14: Prejazd vlaku

Po kliknutí na tlačidlo V (prejazd vlaku): Pri semafore vlaku sa nám zobrazí tlačidlo na spustenie jeho prejazdu a tlačidlo OK na potvrdenie v hornej časti okna (viz obr. 14). Ak si rozmyslíme spustenie vlaku, stačí opäť kliknúť na tlačidlo V.

Informácie o behu simulácie								
Neurónová križovatka	Severná cesta		Východná cesta		Južná cesta		Západná cesta	
	L	S R	L	S R	L	S R	L	S R
Prejdené auta	72	0 72	0 0	72	40	0 72	0 26	72
Prechody	0		0		0		0	
Nehody	0		0		0		0	
Vlak	0		0		0		0	
Čas simulácie	01:16							
Počet prepnutia fáz	3							
Chyby	0							

Informácie o behu simulácie								
Klasická križovatka	Severná cesta		Východná cesta		Južná cesta		Západná cesta	
	L	S R	L	S R	L	S R	L	S R
Prejdené auta	6	26 6	6 0	27 0	6	26 6	6 0	27 0
Prechody	1		1		1		1	
Nehody	0		0		0		0	
Vlak	0		0		0		0	
Čas simulácie	01:16							
Počet prepnutia fáz	3							
Chyby	0							

Obr. 15: Informácie o behu simulácie

Po skončení behu simulácie alebo stlačení tlačidla zastaviť, sa nám zobrazia informácie o simulácii (viz obr. 15). Keď máme spustené obe križovatky, informácie o nich sa zobrazia pre každú zvlášť.

6 Porovnanie

Nový algoritmus na tvorbu dynamického signálneho plánu sme porovnali s klasickým pevným signálnym plánom pomocou informácií o behu z nášho simulátora.

Po desiatich rôzne dlhých simuláciách môžeme povedať, že náš algoritmus využívajúci neurónové siete je efektívnejší než klasický, pretože cez neurónovú križovatku v priemere prešlo viac áut. Môžeme to pripísať tomu, že neurónové siete nechávajú zapnuté pruhy, v ktorých je veľa áut aj niekoľko fáz bez prerušenia, kdežto v prípade klasického režimu je po každej fáze istý časový interval, kedy žiadny pruh nemá signál voľno.

Zistili sme však, že využitie neurónovej siete je efektívnejšie v prípade, že sa na križovatke počas behu simulácie vyskytne aktivovaný prechod, nehoda či prejazd vlaku. Ak ani jedna z týchto udalostí nenastane, teda cez križovatku prechádzajú len autá, stáva sa, že neurónové siete niektorým pruhom nedajú vôbec signál voľno.

Chybovosť siete je uspokojujúca, v niektorých simuláciách dokonca nulová. To nám značí, že naše neurónové siete sa učia celkom správne a teda naše zvolené cvičné dáta boli dostačujúce. Určite sme však nedosiahli najoptimálnejšiu chybovosť, takže v našom algoritme je miesto na zlepšenie.

V tabulke 4 je celkové vyhodnotenie vykonaných simulácií. Ich kompletný prehľad nájdete v prílohe A.

priemerné hodnoty	NEURÓNOVÁ	KLASICKÁ
čas simulácie	11:00	11:00
počet prejdených áut v pruhu	254	145
počet fáz	20	20
počet chýb	3	0
chybovosť	15%	0%

Tabulka 4: Porovnanie

Záver

Podarilo sa nám nájsť algoritmus dynamického prepínania semaforov svetelnej križovatky za použitia neurónových sietí. Tento algoritmus je demonštrovaný aplikáciou simulátoru križovatky a následne porovnaný s klasickým režimom.

Pri porovnaní sme zistili, že algoritmus je počas dopravnej špičky efektívnejší vo väčšine prípadov. Neurónové siete boli vytvorené správne a ich cvičné dáta boli zvolené dobre.

Problém dynamického riadenia križovatky je však komplexný a dá sa k nemu pristupovať rôzne, tak ako aj neurónových sietí existuje veľa druhov použiteľných rôznym spôsobom, preto by sme mohli vytvoriť aj ďalšie algoritmov.

Conclusions

We can conclude that we created an algorithm for dynamic control of traffic light controlled crossroad using neural networks. This algorithm is simulated by Crossroad Simulator application.

The algorithm is more effective compared to the usually used algorithm. Therefore, our neural networks were created correctly and the training data were sufficient.

On the other hand, dynamic crossroad control is very complex problem and along with the variety of neural networks there can be created many other algorithms.

A Prehľad simulácií

Informácie o behu simulácie													Informácie o behu simulácie												
Neurónová križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta			Klasická križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R		L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	308	573	558	101	0	596	0	153	398	0	0	561	124	150	124	122	139	122	124	150	124	122	139	122	
Prechody	0			0			0			0			6			6			6			6			
Nehody	0			0			0			0			0			0			0			0			
Vlak	0			0			0			0			0			0			0			0			
Čas simulácie	10:00												10:00												
Počet prepnutia fází	22												22												
Chyby	0												0												

Obr. 16: Simulácia 1

Informácie o behu simulácie													Informácie o behu simulácie												
Neurónová križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta			Klasická križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R		L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	0	199	415	196	27	415	0	138	330	57	27	315	81	108	81	81	104	81	81	108	81	81	104	81	
Prechody	0			0			0			0			4			4			4			4			
Nehody	0			0			0			0			0			0			0			0			
Vlak	0			0			0			0			0			0			0			0			
Čas simulácie	07:00												07:00												
Počet prepnutia fází	15												15												
Chyby	0												0												

Obr. 17: Simulácia 2

Informácie o behu simulácie												
Neurónová križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	323	27	206	0	107	535	275	0	535	81	189	535
Prechody	0			0			0			0		
Nehody	0			0			0			0		
Vlaky	0			0			0			0		
Čas simulácie	09:00											
Počet prepnutia fází	19											
Chyby	5											

Informácie o behu simulácie												
Klasická križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	108	135	108	108	131	108	108	135	108	108	131	108
Prechody	5			5			5			5		
Nehody	0			0			0			0		
Vlaky	0			0			0			0		
Čas simulácie	09:00											
Počet prepnutia fází	19											
Chyby	0											

Obr. 18: Simulácia 3

Informácie o behu simulácie												
Neurónová križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	305	443	437	170	180	525	374	72	882	27	54	847
Prechody	0			0			0			0		
Nehody	0			0			0			0		
Vlaky	0			0			0			0		
Čas simulácie	16:00											
Počet prepnutia fází	33											
Chyby	1											

Informácie o behu simulácie												
Klasická križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	215	230	215	215	198	215	215	230	215	215	198	215
Prechody	8			9			8			9		
Nehody	0			0			0			0		
Vlaky	0			0			0			0		
Čas simulácie	16:00											
Počet prepnutia fází	33											
Chyby	0											

Obr. 19: Simulácia 4

Informácie o behu simulácie												
Neurónová križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	211	333	868	0	272	428	243	365	682	167	368	547
Prechody	0			0			0			0		
Nehody	0			0			0			0		
Vlaky	0			0			0			0		
Čas simulácie	19:00											
Počet prepnutia fází	39											
Chyby	8											

Informácie o behu simulácie												
Klasická križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	243	267	243	243	262	243	243	267	243	243	262	243
Prechody	10			10			10			10		
Nehody	0			0			0			0		
Vlaky	0			0			0			0		
Čas simulácie	19:00											
Počet prepnutia fází	39											
Chyby	0											

Obr. 20: Simulácia 5

Informácie o behu simulácie												
Neurónová križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	136	25	317	27	111	254	197	27	221	81	80	317
Prechody	1			2			2			0		
Nehody	0			1			0			2		
Vlaky	0			3			0			0		
Čas simulácie	07:00											
Počet prepnutia fází	15											
Chyby	0											

Informácie o behu simulácie												
Klasická križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	84	190	169	54	80	54	54	163	54	54	80	54
Prechody	3			3			3			3		
Nehody	1			0			0			0		
Vlaky	0			3			0			0		
Čas simulácie	07:00											
Počet prepnutia fází	15											
Chyby	0											

Obr. 21: Simulácia 6

Informácie o behu simulácie												
Neurónová križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	461	54	550	54	81	686	314	81	547	107	254	800
Prechody	1			1			1			0		
Nehody	1			1			0			0		
Vlaky	0			1			0			0		
Čas simulácie	15:00											
Počet prepnutia fázi	31											
Chyby	14											

Informácie o behu simulácie												
Klasická križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	187	242	247	162	189	162	217	269	217	162	189	162
Prechody	7			7			7			7		
Nehody	0			0			1			0		
Vlaky	0			2			0			0		
Čas simulácie	15:00											
Počet prepnutia fázi	31											
Chyby	0											

Obr. 22: Simulácia 7

Informácie o behu simulácie												
Neurónová križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	101	168	307	0	57	524	113	201	468	0	54	530
Prechody	1			1			2			2		
Nehody	0			0			0			0		
Vlaky	0			2			0			0		
Čas simulácie	10:00											
Počet prepnutia fázi	21											
Chyby	2											

Informácie o behu simulácie												
Klasická križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	137	191	164	108	126	108	108	165	108	108	126	108
Prechody	5			5			5			5		
Nehody	1			0			0			0		
Vlaky	0			1			0			0		
Čas simulácie	10:00											
Počet prepnutia fázi	21											
Chyby	0											

Obr. 23: Simulácia 8

Informácie o behu simulácie												
Neurónová križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	110	27	260	0	54	224	224	54	194	27	27	257
Prechody	2			2			3			1		
Nehody	0			0			0			0		
Vlak	0			2			0			0		
Čas simulácie	06:00											
Počet prepnutia fáz	13											
Chyby	2											

Informácie o behu simulácie												
Klasická križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	53	139	107	54	54	54	53	139	53	80	83	80
Prechody	2			3			2			3		
Nehody	0			0			0			1		
Vlak	0			2			0			0		
Čas simulácie	06:00											
Počet prepnutia fáz	13											
Chyby	0											

Obr. 24: Simulácia 9

Informácie o behu simulácie												
Neurónová križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	165	464	491	227	138	737	168	225	438	54	0	704
Prechody	1			2			4			2		
Nehody	0			0			0			0		
Vlak	0			0			0			0		
Čas simulácie	14:00											
Počet prepnutia fáz	29											
Chyby	3											

Informácie o behu simulácie												
Klasická križovatka	Severná cesta			Východná cesta			Južná cesta			Západná cesta		
	L	S	R	L	S	R	L	S	R	L	S	R
Prejdené autá	187	216	217	162	188	162	187	212	187	162	188	162
Prechody	7			7			7			7		
Nehody	0			0			0			0		
Vlak	0			1			0			0		
Čas simulácie	14:00											
Počet prepnutia fáz	29											
Chyby	0											

Obr. 25: Simulácia 10

B Obsah přiloženého CD

bin/

Inštalátor CROSSROADSIMULATORINSTALATOR programu CROSSROADSIMULATOR a program CROSSROADSIMULATOR spustitelný přímo z CD. Adresář obsahuje všechny runtime knihnice na bezproblémové spuštění.

doc/

Text práce vo formáte PDF, vytvorený s použitím záväzného štýlu KI PŘF UP v Olomouci pre závěrečné práce, včetně všech příloh, a všechny súbory potrebné na bezproblémové vygenerovanie PDF dokumentu textu (v ZIP archíve), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletné zdrojové texty programu CROSSROADSIMULATOR so všetkými potrebnými zdrojovými textami.

readme.txt

Inštrukcie pre inštaláciu a spuštění programu CROSSROADSIMULATOR, včetně všech požiadavkov na jeho bezproblémový beh.

Literatúra

- [1] HAYKIN, Simon. *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ: Prentice-Hall, 1999. ISBN 0139083855.
- [2] *Navrhování světelných signalizačních zařízení pro řízení silničního provozu: technické podmínky*. Ministerstvo dopravy České republiky, 2015.
Dostupné z: http://www.pjpk.cz/data/USR_001_2_8_TP/TP_81.pdf
- [3] *Návrh signálního programu zadané křižovatky*, České vysoké učení technické.
Dostupné z: [https://k612.fd.cvut.cz/predmety/12ppmk/\(B\)%20Cviceni/12PPMK_Cv05.pdf](https://k612.fd.cvut.cz/predmety/12ppmk/(B)%20Cviceni/12PPMK_Cv05.pdf)
- [4] ROJAS, Raúl. *Neural Networks: A Systematic Introduction*. Springer-Verlag, Berlin, 1996. ISBN 3540605053.
- [5] *Stochastic Gradient Descent – Clearly Explained*
Dostupné z: <https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31>
- [6] *Chapter 5: Training Hidden Units with Back Propagation*.
Dostupné z: <https://web.stanford.edu/group/pdplab/pdphandbook/handbookch6.html>
- [7] *Windows Presentation Foundation documentation*
Dostupné z: <https://docs.microsoft.com/en-us/dotnet/desktop/wpf/?view=netdesktop-5.0>