

# STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

## Využití neuronové sítě pro fotbal robotů

Kateřina Kolaříková  
Moravskoslezský kraj

Ostrava 2019

# STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

**Využití neuronové sítě  
pro fotbal robotů**

**Artificial intelligence  
for robot soccer**

**Autor:** Kateřina Kolaříková

**Škola:** Gymnázium Olgy Havlové, Marie Majerové 1691/4,  
708 00 Ostrava-Poruba

**Kraj:** Moravskoslezský kraj

**Konzultant:** Ing. Jan Martinovič, Ph.D.

Ostrava 2019

## **Prohlášení**

Prohlašuji, že jsem svou práci SOČ vypracovala samostatně a použila jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů. Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Ostravě dne 14. 3. 2019 .....

Kateřina Kolaříková

## **Poděkování**

Ráda bych na tomto místě poděkovala všem, kteří mi pomohli při tvorbě práce. Zejména děkuji Ing. Janu Martinovičovi, Ph.D. za odborné vedení práce. Dále bych ráda poděkovala Ing. Václavu Svatoňovi, Ph.D. a Sebastiánovi Husárovi za seznámení se simulátorem fotbalu robotů. Mé díky patří také Ing. Jakubovi Beránkovi za konzultace v oblasti neuronových sítí.

## Anotace

V dnešní době se stále více můžeme setkat s pojmem umělá inteligence a jejím využitím v různých oblastech průmyslu i běžného života. Mezi tyto oblasti patří také sportovní datová analýza. Právě aplikací jedné z metod umělé inteligence do oblasti sportu se zabývá má práce. Jejím cílem je využití neuronových sítí v rámci simulátoru fotbalu robotů. Cílem bylo vytvoření neuronové sítě schopné rozhodovat o pohybu hráčů a její otestování v simulované hře. K vývoji a testování byl využit programovací jazyk Python a testovacím prostředím byl simulátor fotbalu robotů RobotSoccerSimulator vyvíjený na Vysoké škole báňské – Technické univerzitě Ostrava, konkrétně na národním superpočítacovém centru IT4Innovations.

## Klíčová slova

Umělá inteligence; neuronová síť; fotbal robotů; Python

## Annotation

Nowadays, we can increasingly meet the concept of artificial intelligence and its use in various areas of industry and everyday life. These areas also include sports data analysis. My work is dedicated to the application of one AI method to the field of sport. Its goal is to use neural networks within the robot football simulation. The aim was to create a neural network capable to make decisions about the players movement and testing them in a simulated game. The Python programming language was used to develop code. Robot Soccer Simulator is developed at the Technical University of Ostrava, specifically at the national supercomputer center IT4Innovations.

## Keywords

Artificial Intelligence; Robot Soccer; Nerural Network; Python

# **Obsah**

<b>1</b>	<b>Úvod</b>	<b>7</b>
<b>2</b>	<b>Fotbal robotů</b>	<b>7</b>
2.1	Popis simulátoru . . . . .	7
2.2	Strategie . . . . .	9
2.3	Taktiky . . . . .	10
2.4	Herní logy . . . . .	10
2.4.1	Popis logu . . . . .	10
<b>3</b>	<b>Neuronová síť</b>	<b>11</b>
3.1	Biologická neuronová síť . . . . .	11
3.2	Neuron . . . . .	12
3.3	Struktura sítě . . . . .	13
3.4	Proces učení . . . . .	14
<b>4</b>	<b>Analýza dat z her fotbalu robotů</b>	<b>14</b>
4.1	Analýza odehrané hry pomocí teplotních map . . . . .	16
4.2	Sehrání zápasů . . . . .	17
4.2.1	Vyhodnocení zápasů . . . . .	17
<b>5</b>	<b>Využití neuronové sítě pro fotbal robotů</b>	<b>19</b>
5.1	Noc vědců . . . . .	19
5.2	Vstupní data pro neuronovou síť . . . . .	20
5.3	Proces trénování sítě . . . . .	21
5.3.1	Komunikace simulátoru se sítí . . . . .	24
5.4	Ověření funkčnosti sítě . . . . .	25
5.4.1	Vznik testovací sítě . . . . .	25
5.4.2	Test sítě . . . . .	26
5.5	Zápasy neuronové sítě . . . . .	27
5.5.1	Vyhodnocení zápasů . . . . .	28
5.5.2	Porovnání různých vstupních dat . . . . .	28
5.5.3	Vliv dat na výsledky zápasů . . . . .	29
<b>6</b>	<b>Závěr</b>	<b>30</b>
<b>7</b>	<b>Seznam referencí</b>	<b>31</b>
<b>8</b>	<b>Seznam obrázků</b>	<b>32</b>
<b>9</b>	<b>Seznam příloh</b>	<b>33</b>

# 1 Úvod

Umělá inteligence nabyla v posledních letech neskutečné popularity. Setkáváme se s ní v rámci běžného života, jelikož je například aplikována v mobilních asistentech jako je Siri<sup>1</sup> nebo Google Assistant<sup>2</sup>, či v rámci fotoaparátů některých chytrých telefonů a v mnoha dalších oblastech. Zároveň je jednou z atraktivních oblastí vědy a výzkumu a neustále se proto pracuje na jejím vývoji. Jedním z nejčastěji využívaných modelů v oblasti umělé inteligence jsou neuronové sítě [1]. Ty byly v nedávné době využity mimo jiné pro hraní her. Například ve hře Dota II umělá inteligence<sup>3</sup> dokázala opakovaně porazit tým složený z nejlepších hráčů na světě.

V rámci této práce jsem se rozhodla otestovat využití neuronových sítí jako modelu umělé inteligence pro hraní virtuálního fotbalu robotů. V první části práce se zabývám popisem simulátoru fotbalu robotů (Sekce 2) a neuronových sítí (Sekce 3). V druhé části popisuji analýzu dat vytvářených herním simulátorem (Sekce 4) a proces tvorby neuronové sítě pro hraní fotbalu robotů (Sekce 5). Závěry práce jsou uvedeny v Sekci 6.

## 2 Fotbal robotů

Fotbal robotů je hra, ve které dva týmy robotů soupeří ve zjednodušené verzi klasického fotbalu. Existují mezinárodní turnaje, ve kterých měří síly robotické týmy, např. RoboCupSoccer<sup>4</sup>. Pro hru jsou důležité hned dvě části. Jednak sestavit hardware tvořící hráče, ale také naimplementovat software, tj. "myšlení" robota. Já jsem se zaměřila na druhou část, pro kterou jsem se rozhodla využít neuronové sítě. Aby se sítěmi šlo rychle experimentovat, používám virtuální simulátor fotbalu robotů, který byl vyvinut na Vysoké škole báňské<sup>5</sup>.

### 2.1 Popis simulátoru

Simulátor zprostředkovává fotbalové utkání dvou týmů robotů. Každý zápas trvá dvě minuty a výhercem je tým, který vstřelí největší počet gólů. V případě, že oba dva týmy mají stejný počet gólů, tak zápas končí remízou. Při popisu týmů se užívá rozdelení na levý a pravý tým. Každý tým čítá pět

---

<sup>1</sup><https://www.apple.com/siri/>

<sup>2</sup><https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html>

<sup>3</sup><https://blog.openai.com/openai-five/>

<sup>4</sup><https://www.robocup.org/leagues/7/>

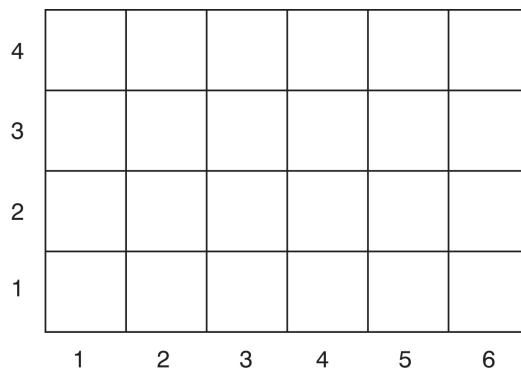
<sup>5</sup><https://github.com/It4innovations/RobotSoccer/>

hráčů, z nichž jeden představuje brankáře. Kromě dvou týmů se na hřišti pohybuje také míč. Ukázka herní situace je vidět na Obrázku 1.



Obrázek 1: Ukázka simulátoru

Herní hřiště je reprezentováno gridem o rozměrech  $6 \times 4$ . To znamená, že herní plán se skládá z celkem 24 polí. Každému z polí náleží souřadnice  $(X,Y)$ , kde X určuje horizontální souřadnici a Y určuje vertikální souřadnici. Levému dolnímu rohu je přiřazena souřadnice  $(1,1)$ , pravému hornímu rohu pak náleží souřadnice  $(6,4)$ . Ukázku rozložení hřiště si lze prohlédnout na Obrázku 2. Každý objekt na hřišti (hráč nebo míč) se v daný moment nachází právě na jednom poli herního plánu.



Obrázek 2: Ukázka gridu

## 2.2 Strategie

Chování a postavení robotů je řízeno strategiemi. Strategie ovládá chování hráčů jednoho týmu. Skládá se ze sady pravidel, které jsou reprezentovány textovým souborem. Každé pravidlo obsahuje popis herní situace a seznam pozic, kam se mají přesunout jednotliví hráči týmu v případě aplikace daného pravidla. Ukázku pravidla si lze prohlédnout na Obrázku 3. První řádek pravidla udává jeho název a pořadí. Následující tři řádky popisují herní situaci, při které se má pravidlo použít. Řádek *Mine* zaznamenává pozice hráčů týmu, který strategie ovládá. Následující řádek *Oppnt* obsahuje pozice soupeřova týmu a *Ball* reprezentuje pozici míče. Parametr *Move* říká, kam se mají hráči týmu přesunout při aplikaci pravidla.

Simulátor během hry opakovaně vybírá pravidlo, jehož popis herní situace je nejpodobnější aktuálnímu dění na hřišti. To znamená, že při vytváření pravidel nemusíme pokrýt každou možnou herní situaci. Vybrané pravidlo je poté aplikováno, což znamená, že hráči týmu se začnou přesouvat na pozice uvedené v instrukcích *Move*. Výběr pravidla probíhá každých 20 ms.

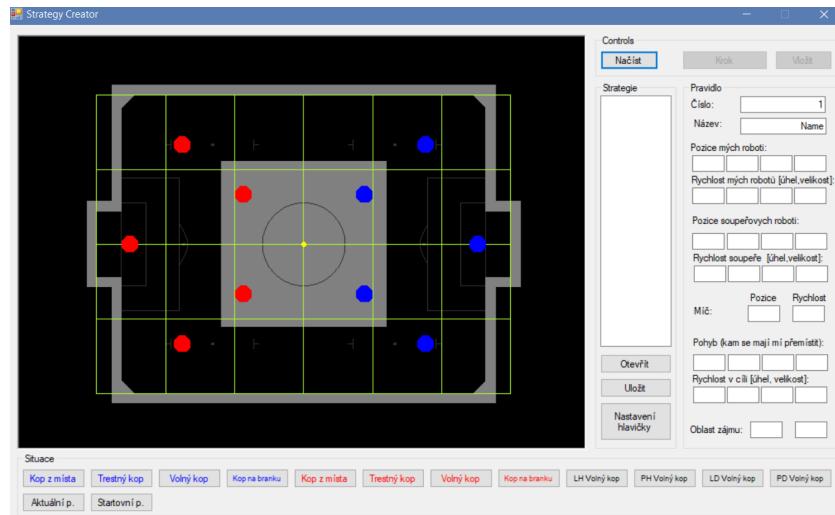
Využití hrubého gridu (6 x 4) nám usnadňuje vytváření strategií. Pokud bychom použili jemný grid vzniklo by příliš velké množství situací, které by bylo pravidly nutné pokrýt.

```
.Rule 1 o def
.Mine 3,2 3,3 2,1 2,4
.Oppnt4,2 3,3 5,1 5,4
.Ball 3,3
.Move 2,3 3,3 2,1 2,4
```

Obrázek 3: Ukázka pravidla

Strategie lze vytvářet pomocí programu StrategyCreator viz Obrázek 4. Jedná se o rozhraní, které zjednodušuje tvorbu pravidel, aby je nebylo nutné vytvářet ručně. Při tvorbě je nutné navolit herní situaci a pohyb hráčů týmu. Nově vzniklá pravidla lze dělit dle jejich funkce, např. na defenzivní (obranná) a ofenzivní (útočná).

Je patrné, že se samostatnými strategiemi si nevystačíme. Občas některým robotům potřebujeme přiřadit speciální funkce, např. brankář nebo útočník, proto byly do hry zařazeny tzv. taktiky.



Obrázek 4: Ukázka StrategyCreatoru

## 2.3 Taktiky

Taktiku můžeme přiřadit konkrétnímu robotovi v případě, že chceme, aby plnil speciální úlohy, které nejsou v rámci strategií obsaženy. Jednou z nejtypičtějších taktik je brankář, jehož hlavním úkolem je pohyb okolo brány a její obrana. Jiné taktiky se naopak dynamicky mění během hry, např. "útočník", který je vždy přiřazen robotovi nacházející se nejbližše míče.

Taktiky jsou tedy nadřazený strategií. To znamená, že robot, kterému byla přiřazena taktika se bude dočasně řídit právě jí, nikoliv instrukcemi uvedenými ve strategii.

## 2.4 Herní logy

Po každém odehraném zápase je simulátorem vygenerován tzv. herní log. Log je textový soubor, který nám umožňuje se zpětně podívat na průběh hry. Zaznamenává jak použitá pravidla, tak samotný průběh hry.

### 2.4.1 Popis logu

Samotný log je tvořen záznamy, které vždy obsahují řadu dvaceti čísel, jak je vidět na přiloženém Obrázku 5. V prvním sloupci nalezneme číslo pravidla užitého levým týmem. V druhém sloupci je uvedeno pravidlo vybrané pravým týmem. Následující dvojice čísel popisuje míč. A to tak, že první číslo je X-ová a druhé Y-ová souřadnice. Následujících deset čísel přináší informaci o rozložení robotů levého týmu na hřišti. Jednoho robota vždy reprezentuje

dvojice čísel zaznamenaná souřadnicemi v pořadí X,Y. Obdobně to platí pro následujících deset čísel, které představují pravý tým. Následuje aktuální skóre a čas, ve kterém byl daný řádek logu zapsán.

	použitá pravidla		souřadnice míče		pozice levých robotů											
	A	B	C	D	E	F	G	H	I	J	K	L	M	N		
1	leftRule	rightRule	ball.x	ball.y	lr0.x	lr0.y	lr1.x	lr1.y	lr2.x	lr2.y	lr3.x	lr3.y	lr4.x	lr4.y		
2	6	4	4	3	2	2	3	3	3	2	2	3	1	2		
3	6	6	4	3	2	2	3	3	3	2	2	3	1	2		

	pozice levých robotů															pozice pravých robotů		
	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	score	time	control		
1	rr0.x	rr0.y	rr1.x	rr1.y	rr2.x	rr2.y	rr3.x	rr3.y	rr4.x	rr4.y								
2	5	2	4	3	4	2	5	3	6	2	0:00	2:00	0					
3	5	2	4	3	4	2	5	3	6	2	0:00	1:59	0					

Obrázek 5: Ukázka herního logu

K zápisu dochází co 20 milisekund. To znamená, že během jedné dvouminutové hry vznikne zhruba 6 tisíc řádků logu.

### 3 Neuronová síť

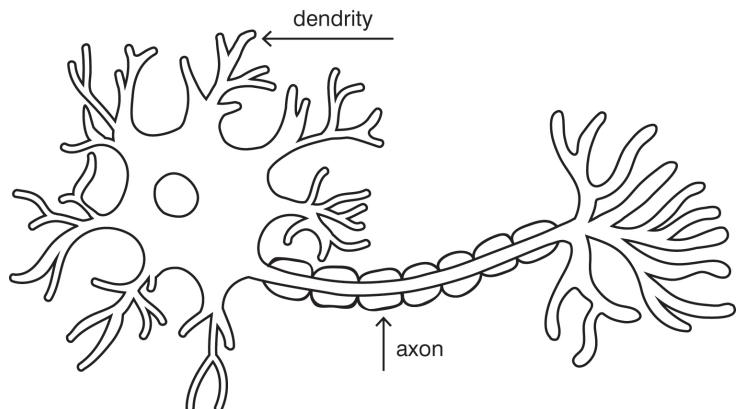
Umělá neuronová síť je jeden z výpočetních modelů umělé inteligence, který je inspirován fungováním biologických neuronových sítí [1].

#### 3.1 Biologická neuronová síť

Neuronová síť přijímá ze svého okolí velké množství informací. Tyto informace tělo převádí na slabé elektrické impulsy, tzn. vzruchy. Základní složkou sítě jsou neurony viz Obrázek 6, které zajišťují další šíření vzruchu.

Samotný neuron se skládá z těla a výběžků (tzn. dendrity a axon), kdy dendritů může být až několik tisíc, ale axon je jen jeden.

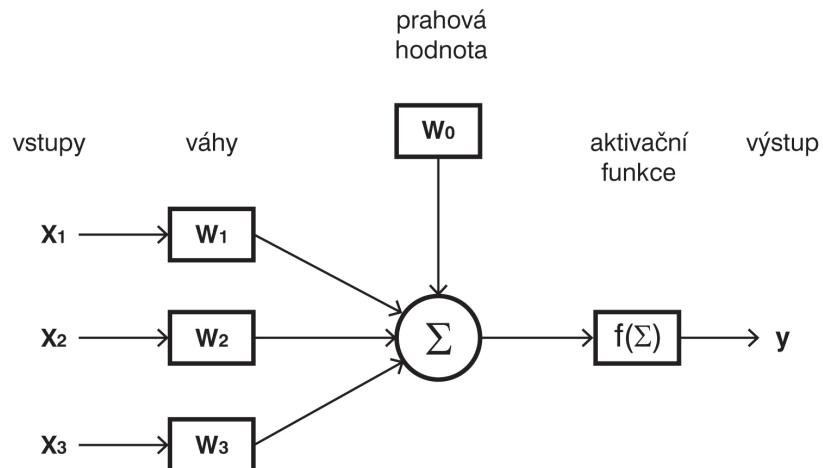
Úkolem dendritů je přijmout vzruch. Pokud je tento signál dostatečně silný, dojde k vyvolání vzruchu, který se prostřednictvím axonu šíří dále. Propojení jednotlivých neuronů do rozsáhlé sítě umožňuje jejich vzájemnou komunikaci a je základním stavebním kamenem inteligence.



Obrázek 6: Nákres biologického neuronu

### 3.2 Neuron

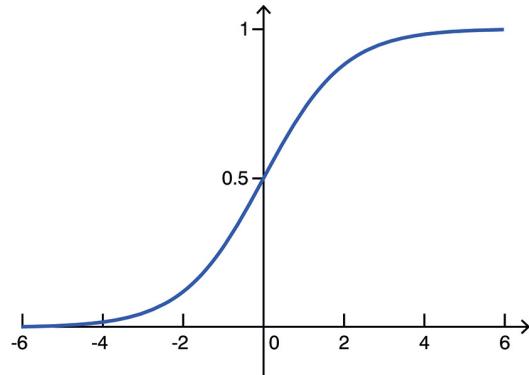
Základní stavební jednotkou umělé neuronové sítě je, stejně jako u té biologické, neuron viz Obrázek 7.



Obrázek 7: Model umělého neuronu

Neuron má více vstupů, ale vždy jen jeden výstup. Každý ze vstupů má přidělenou určitou váhu, která udává důležitost vstupu. Jakmile přijme daný vstup signál, je vynásoben právě touto váhou. Tyto součiny jsou poté sečteny a dostaneme jedno konečné číslo. Dále se používá tzv. bias neboli prahová hodnota. Představuje váhu, která má fixně nastavenou velikost vstupu a má tak vliv na velikost výsledného součtu. Toto sečtené číslo je následně vyhodnoceno aktivací funkci, která na základě jeho velikosti rozhoduje o tom, zda vyšle impuls do další vrstvy či nikoliv. Tato funkce může mít různé podoby.

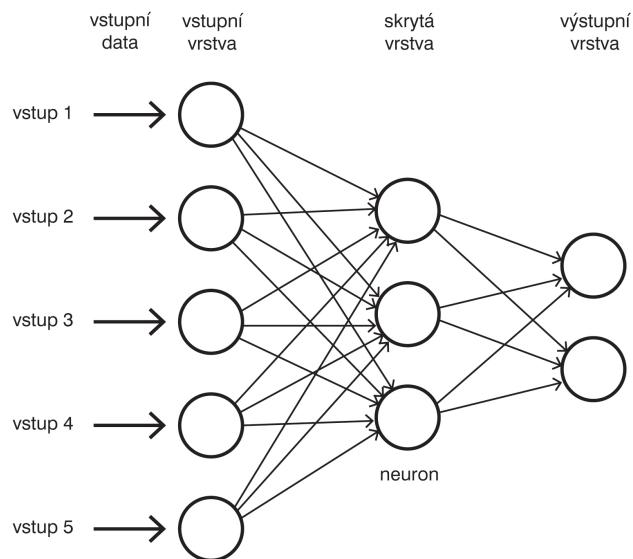
Jednou z nejčastěji využívaných je funkce sigmoid. Průběh této funkce si lze prohlédnout na Obrázku 8.



Obrázek 8: Funkce sigmoid

### 3.3 Struktura sítě

Jednotlivé neurony se spojují do vrstev. První vrstva se nazývá vstupní a jejím úkolem je příjem vstupních informací pro síť, které posílá dále.



Obrázek 9: Struktura neuronové sítě

Za ní následují skryté vrstvy. V nich probíhá hlavní práce neuronové sítě. Každá vrstva se skládá z neuronů, které jsou propojeny s neuronami další vrstvy. Neuron z předešlé vrstvy je spojen s každým z neuronů z vrstvy

následující, viz Obrázek 9. Těmto spojům náleží váhy. Informace, které předá vstupní vrstva, jsou vyhodnoceny první skrytou vrstvou. Získané informace jsou předány následující vrstvě, a takto se vše opakuje pro každou ze skrytých vrstev. Kolik bude skrytých vrstev a z kolika neuronů se budou skládat je individuální, záleží na konkrétním problému, který má sít za úkol vyřešit.

Poslední typ vrstvy se nazývá výstupní. Ta má tolik neuronů, kolik je možných výstupů. Je mnoho možností, jak můžeme čísla na výstupu interpretovat, závisí to především na typu řešeného problému. Jedním z častých výkladů je procentuální pravděpodobnost správnosti daného výstupu.

### 3.4 Proces učení

Existuje mnoho způsobů, jakým může probíhat učení neuronové sítě [3, 2]. Jedním z nich je tzv. učení s učitelem. Během něj dochází k porovnávání výstupu sítě s očekávaným výstupem a na základě těchto srovnání dojde k úpravě vah uvnitř neuronové sítě.

Proces učení sítě se skládá ze dvou částí, z trénování a testování. Nejdříve je ovšem třeba získat data, na kterých budeme sít učit a testovat. Ty se skládají ze dvou složek, *vstupní* a *výstupní*. Ve složce *vstupní* se nachází stejná data, která odpovídají požadovaným *vstupním* datům pro sít. Složka *výstupní* pak nese požadovaný výstup.

Tuto množinu dat rozdělíme na dvě části, trénovací a testovací.

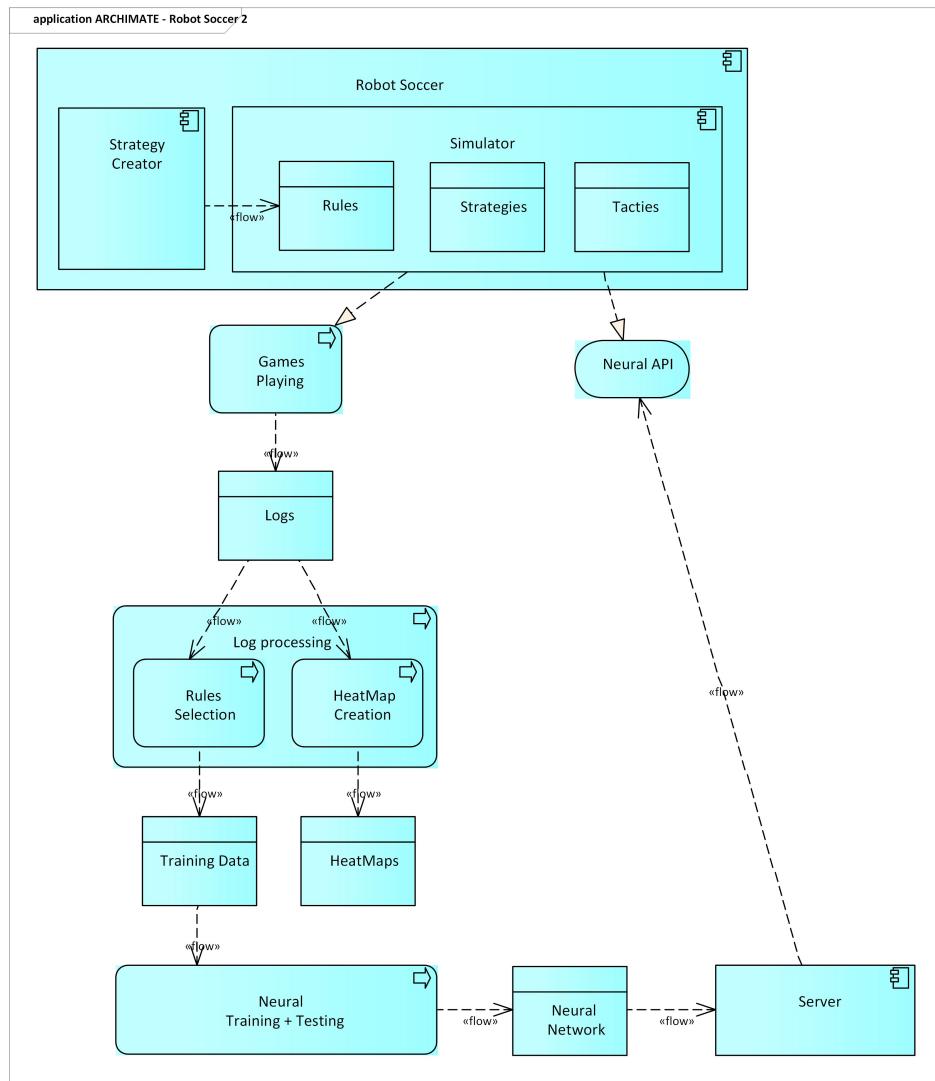
Během procesu je síti předložena trénovací množina dat, ze kterých se sít učí reagovat na jednotlivé vstupy.

Další fází je testování, kdy dochází k porovnávání výstupu sítě s očekávaným výstupem, který je obsažen ve složce *výstupní*. U těchto testovacích dat počítáme chybu sítě, tzn. kolikrát a jak moc se během testu spletla. Cílem trénování je naučit síť, aby zobecnila trénovací data a dokázala řešit problém i pro vstupy, které ještě nikdy neviděla. Úspěšnost (byť stoprocentní) na trénovací sadě nemusí znamenat, že je síť správně natrénovaná. Je možné, že si síť jednoduše zapamatovala celou trénovací sadu. Proto je třeba úspěšnost testovat na datech, které síť nikdy neviděla (tzv. testovací množina). Tak lze otestovat funkčnost sítě v podmínkách, ve kterých má být používána.

## 4 Analýza dat z her fotbalu robotů

Proces využití umělé inteligence pro simulátor fotbalu robotů lze vidět na Obrázku 10. Nejdříve byly za pomocí StrategyCreatoru vytvořeny strategie. Mezi strategiemi se prostřednictvím simulátoru nechaly sehrát zápasy. Z těchto zápasů vznikly zápisu v podobě logů. Logy byly zpracovány dvěma

způsoby. Prvním způsobem byla jejich analýza pomocí teplotních map viz Sekce 4.1. Druhým využitím bylo získání dat pro vznik modelu neuronové sítě. Vzniklý model sítě poté prostřednictvím serveru komunikoval se simulátorem a ovládal chování jednoho z týmu robotů.



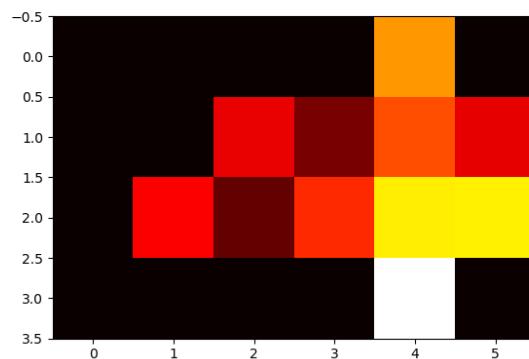
Obrázek 10: Struktura práce

Poté, co jsme si představil celý proces, je třeba se zamyslet nad jednotlivými akcemi v něm, nejdříve nad analýzou dat. Jednoznačným parametrem určujícím úspěšnost daného týmu je skóre. Tým, který vyhrál, lze považovat za kvalitnější. Abychom ovšem dokázali rozpoznat, co je pro daný tým vhodné a vede k jeho výhře, měli bychom se zaměřit i na další části herních logů.

Jedním z možných přístupů je zkoumání využitých pravidel. Mohli bychom je například začít dělit na úspěšná a neúspěšná v závislosti na tom, zda po nich došlo k vstřelení či dostání gólu. Druhou možností je analýza hry na základě pozic jednotlivých herních objektů. V této práci byla zvolena právě tato druhá varianta, protože dění na hřišti popisuje daleko detailněji než pravidlo. Jako data k analýze byly tedy vybrány právě pozice všech objektů na hřišti. Nástrojem analýzy byly teplotní mapy viz Sekce 4.1. K jejich vytvoření bylo napsáno několik skriptů v jazyce Python. Vždy vznikly celkem tři mapy. Zobrazovaly četnosti výskytů, jedna pro míč, druhá pro roboty pravého týmu a třetí pro roboty levého týmu. Pro každé pole herního plánu byl vytvořen součet výskytů daného objektu (míče, robotů levého týmu či robotů pravého týmu). Tyto číselné hodnoty byly poté převedeny na jednotlivé teplotní mapy. Cílem této analýzy bylo zjistit, jak se projeví výběr strategie na pohybu hráčů a vývoji hry.

## 4.1 Analýza odehrané hry pomocí teplotních map

Jak již bylo výše zmíněno, k analýze byly využity teplotní mapy viz. Obrázek 11. Teplotní mapa je grafické zobrazení dat za pomoci barevného spektra. Spektrum je tvořeno omezeným výběrem barev, bílou počínaje, přes světle žlutou, oranžovou, tmavě červenou až po černou. Bude - li nejvyšší hodnotě, kterou chceme zobrazit přiřazena nejsvětlejší barva, bude spolu se snižujícími hodnotami dat barva tmavnout, až dosáhne nejnižší hodnoty a nejtmavší barvy.



Obrázek 11: Ukázka teplotní mapy

Teplotní mapa byla vybrána vzhledem k tomu, že cílem bylo zobrazení

herní plochy. Můžeme ji promítout na celé hřiště a dá se tedy velmi dobře představit vývoj pohybu robotů na herním poli.

## 4.2 Sehrání zápasů

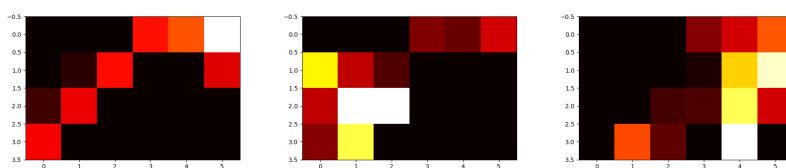
Dalším krokem bylo získání logů k analýze. Za tímto účelem došlo k vytvoření dvou rozdílných strategií. Jedna měla čistě defenzivní zaměření, to znamená, že jediným smyslem pravidel byla obrana brány. Druhá strategie měla ofenzivní charakter, takže hlavním účelem bylo pohybovat se před bránou soupeře a snažit se dát gól. Na základě této dělby byly strategie pojmenovány *Defenziva* a *Ofenziva* a byly tvořeny 20 pravidly.

Sehrály se tyto kombinace, *Ofenziva* vs. *Ofenziva*, *Defenziva* vs. *Defenziva*, *Defenziva* vs. *Ofenziva*. Každá dvojice hrála vždy dvacet zápasů. Po dohrání zápasů byly vygenerovány teplotní mapy vždy po 500 řádcích logu. Každá série řádků se počítala od nuly, tzn. každá teplotní mapa reprezentovala součet pozic za samostatných 500 řádků.

### 4.2.1 Vyhodnocení zápasů

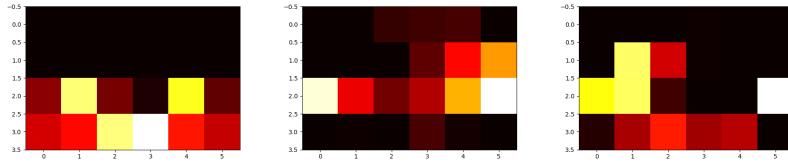
Při prohlížení map bylo důležité především zjistit, zda jejich výstup odpovídá smyslu, v jakém byly tyto strategie psány.

V případě defenzivního týmu byla pravidla vymýšlena tak, aby se co nejvíce pohybu robotů odehrálo před jejich vlastní branou. Podíváme-li se na Obrázek 12, první mapa reprezentuje míč, druhá roboty levého týmu a třetí roboty pravého týmu, uvidíme, že oba týmy se skutečně držely před svou branou. Drobný výskyt před branou soupeře byl způsoben taktikou útočníka, viz Sekce 2.3. Ta totiž způsobila, že robot v blízkosti míče se vydal do útoku před branou soupeře.



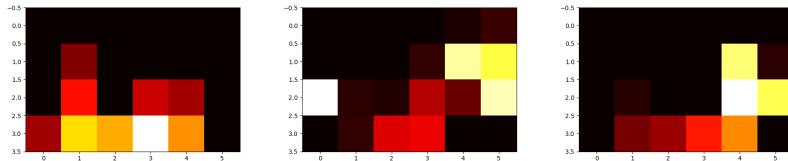
Obrázek 12: Defenziva vs. Defenziva

Roboti reprezentující ofenzivní tým se taktéž nejvíce vyskytovali před branou, nikoliv však před svou vlastní, ale před soupeřovou, jak nám dokládá Obrázek 13.



Obrázek 13: Ofenziva vs. Ofenziva

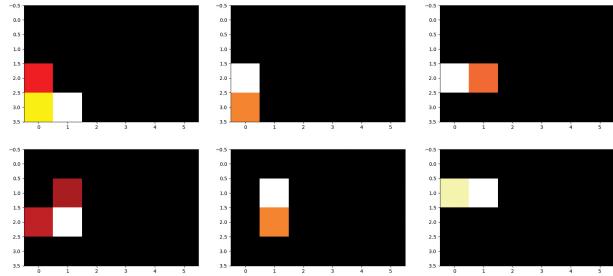
Díky tomuto testu došlo k ověření funkčnosti jednotlivých strategií. Pozorovat jejich chování bylo zajímavé i na živo, tedy sledováním jednotlivých zápasů. Za zmínku stojí například zápasy mezi *Ofenzivním* a *Defenzivním* týmem viz. Obrázek 14, jelikož se nejlépe projevila rozdílnost jednotlivých strategií. Hráči ofenzivního týmu museli protlačit míč přes hustou obranu *Defenzivních*. O góly defenzivního týmu se naopak postarala taktika útočníka. Což znamená, že dostal-li se míč do blízkosti některého z robotů, vyrazil s ním k soupeřově bráně, kde proti němu stál již jen brankář. Zbytek týmu byl totiž v útočných pozicích na druhé straně hřiště.



Obrázek 14: Ofenziva vs. Defenziva

Vytvářet teplotní mapy po 500 řádcích ovšem neumožňuje sledovat průběh hry detailněji, proto bylo využito Sliding window. Jedná se o metodu, která slouží ke kontrole dat. Každá sada dat není vyhodnocována samostatně, ale s určitým prolnutím se sadou předchozí.

Původní skripty pro tvorbu teplotních map byly upraveny, aby vždy polovina řádků byla užita i v mapě předchozí např. 0-20, 10-30, 20-40. Toto zobrazení umožnilo sledovat průběžný vývoj hry, jelikož při prohlížení map za sebou byl patrný pohyb hráčů i míče. Obrázek 15 zobrazuje šest za sebou jdoucích teplotních map ukazujících výskyt míče. První fáze pohybu je vidět v levém horním rohu a dále teplotní mapy čteme po řádcích.



Obrázek 15: Ukázka pohybu míče

## 5 Využití neuronové sítě pro fotbal robotů

Díky seznámení se simulátorem a herními logy již bylo možné přistoupit k samotné tvorbě sítě. V předchozí části práce byly strategie voleny buď čistě ofenzivní nebo čistě defenzivní. Abychom zajistili co nejlepší naučení neuronové sítě, je třeba ji naučit reagovat na co nejvíce možných situací, tedy jak defenzivních, tak ofenzivních. K tomu byla využita Noc vědců.

### 5.1 Noc vědců

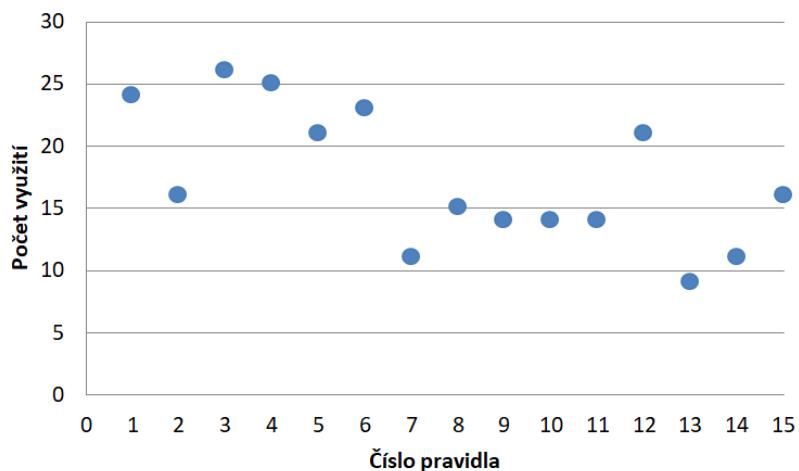
V rámci akce Noc vědců, do kterého se zapojilo také IT4Innovations národní superpočítacové centrum, jsme získali sadu logů.



Obrázek 16: Ukázka z noci vědců

Pro tuto událost bylo připraveno 15 pravidel. Ty byly předloženy návštěvníkům, každý si mohl čtyři vybrat. Z těchto čtyř pravidel byla vytvořena strategie. Strategie dvou účastníků se nechávaly hrát proti sobě. K sehrání zápasů byl využit stejný simulátor, který byl využíván v této práci. Z těchto zápasů vzniklo 60 logů. Oněch 15 pravidel bylo tvořeno pěti defenzivními, pěti ofenzivními a pěti speciálními pravidly. Speciální poslaly roboty na místa, která neměla ani ofenzivní ani defenzivní charakter, jako je například střed hřiště. Tato variabilita pravidel umožňovala lidem vytvářet různé typy strategií a herních situací.

Jak vyplývá z grafu na Obrázku 17, byla využita rozličná pravidla v rozličné míře. To zajistilo velké množství rozdílných herních situací, ze kterých se sít mohla učit. Největší oblibě se těšila pravidla 1, 3 a 4, které měly ofenzivní charakter. Z defenzivních pravidel, kterým naleží čísla 5-10 bylo nejvyužívanější pravidlo 6. Speciálním pravidlům odpovídají čísla 11 až 15. Mezi těmito pravidly taky nalezneme nejméně využité ze všech, pravidlo číslo 13.



Obrázek 17: Graf využití pravidel

Zároveň byl upraven samotný simulátor a to tak, aby byl zmenšen vliv taktiky útočníka. Toho se dosáhlo snížením vzdálenosti, která musela být mezi míčem a robotem. Tento krok vznikl z důvodu, aby se snížil vliv taktik a zvýšil se vliv samotných strategií.

## 5.2 Vstupní data pro neuronovou síť

Abychom využili neuronovou síť pro fotbal robotů, je třeba se v prvé řadě zamyslet nad vstupními daty. Vzhledem k tomu, že díky logům byl k dispozici

přesný popis pozic všech objektů na hřišti, jako vstupní data byly zvoleny právě tyto informace. Výstupní složkou bylo číslo pravidla, které odpovídalo dané herní situaci. A to z důvodu, že cílem bylo, aby síť rozhodovala, kam se mají roboti týmu, za který hrála, přemístit. Díky číslu pravidla, byly k dispozici i instrukce Move, uvedené v příslušné strategii.

### 5.3 Proces trénování sítě

K natrénování neuronové sítě byl využit jazyk Python a knihovna TensorFlow<sup>6</sup>. Jedná se o open-source knihovnu od Googlu, která je dnes jedním z nejpoužívanějších nástrojů pro trénování neuronových sítí. Druhou využitou open-source knihovnou byl Keras<sup>7</sup>.

Pro předání herní situace neuronové sítě byly využity vícerozměrné pole. Celý princip si vysvětlíme na zjednodušené herní situaci. Zmenšíme herní pole na rozměry ( 2x 2) a snížíme počet hráčů. Na hřišti budou pouze tři objekty: míč, jeden hráč pravého a jeden hráč levého týmu. Celý herní plán si znázorníme pomocí vícerozměrného pole, kdy každé jednotlivé pole v něm představuje jeden díl herního plánu viz. Obrázek 18

(1,2)	(2,2)
(1,1)	(2,1)

$$\begin{matrix} (1,1) & (2,1) & (1,2) & (2,2) \\ [[0,0,0] & [0,0,0] & [0,0,0] & [0,0,0]] \end{matrix}$$

Obrázek 18: Znázornění herního plánu pomocí vícerozměrných polí

Přístup k těmto polím byl získán pomocí jednoduchého vzorce  $(y-1)*z+x-1$ , kdy  $y$  představuje  $Y$ -novou souřadnici objektu,  $x$  představuje  $X$ -ovou souřadnici objektu a  $z$  velikost gridu na  $x$ -ové ose. Díky tomu jsme

---

<sup>6</sup><https://www.tensorflow.org>

<sup>7</sup><https://keras.io>

převedli souřadnice  $(X, Y)$  každého z objektů na hřišti na index. Pomocí tohoto indexu jsme získali přístup k jednotlivým polím ve vícerozměrném poli. Za míč a každého z robotů byla do pole přičtena jednička. Abychom odlišili, zda se jedná o míč či některého z robotů, záleželo na pozici, na kterou byla zapsána. Pokud se jednalo o míč, byla zapsána na nultou pozici, pokud o robota levého týmu tak na první pozici a pokud o robota pravého týmu tak na pozici druhou, viz Obrázek 19.

		pravý
(1,2)		(2,2)
	míč	levý
(1,1)		(2,1)

$$\begin{matrix} (1,1) & (2,1) & (1,2) & (2,2) \\ [[1,0,0] & [0,1,0] & [0,0,0] & [0,0,1]] \end{matrix}$$

Obrázek 19: Převod pozic objektů

Úplně stejný princip byl využit, abychom síti předali informace na hřišti o rozměrech (6 x 4), po kterém se pohybovalo deset robotů a míč. Abychom mohli síť vytvořit, bylo kromě získání dat nutné určit její strukturu.

V této práci byla zvolena tato struktura sítě - vstupní vrstva, po ní následuje jedna skrytá vrstva s 512 neurony a poslední je výstupní vrstva s 15 neurony. Jeden neuron výstupní vrstvy reprezentuje jedno pravidlo. Proběhly experimenty s různými hodnotami ve struktuře, výsledky byly ovšem téměř shodné. Ukázku kódu je vidět níže.

```
def neural_network(x_train,y_train,x_test,y_test):

    model = tf.keras.models.Sequential([
        Flatten(input_shape=x_train[0].shape),
        Dense(512, activation=tf.nn.relu), # "husta vrstva", 512-neurony
        Dense(15, activation=tf.nn.softmax) #activation - aktivacni funkce
        # neuron pro každou číslici, hodnota udává pravděpodobnost
    ])
```

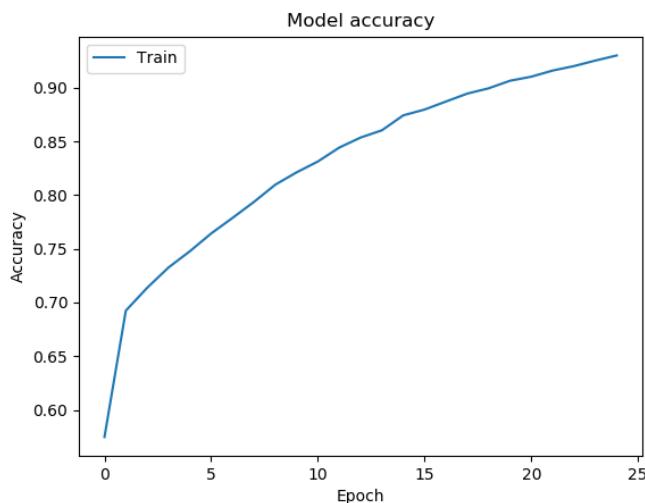
```

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

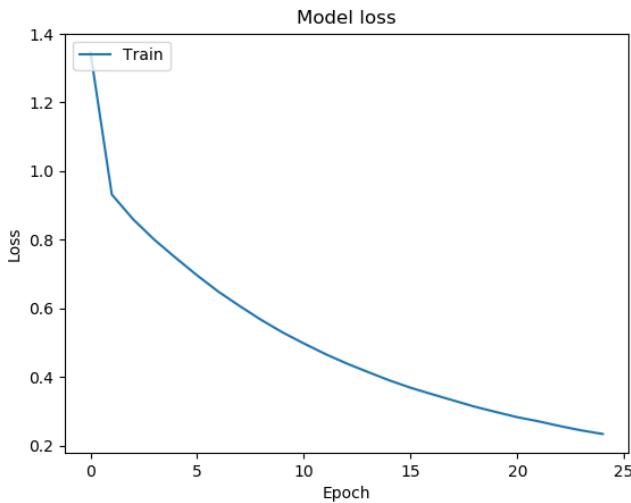
history = model.fit(x_train, y_train, epochs=25) #fit == train

```

Kvalita procesu učení sítě, byla zachycena pomocí dvou grafů. Graf na Obrázku 20 zachycuje změnu přesnosti se zvyšujícím se počtem epoch. Epocha značí jeden časový krok učení, kdy síť byla předložena jak vstupní tak výstupní data. Můžeme vidět, že čím bylo epoch více, tím přesněji síť pracovala. Graf na Obrázku 21 se opět odkazuje na počet epoch, tentokrát ukazuje průměrnou chybu na výstupu, kdy s rostoucím počtem epoch se chyba sítě zmenšovala.



Obrázek 20: Grafy přesnosti sítě na trénovací množině

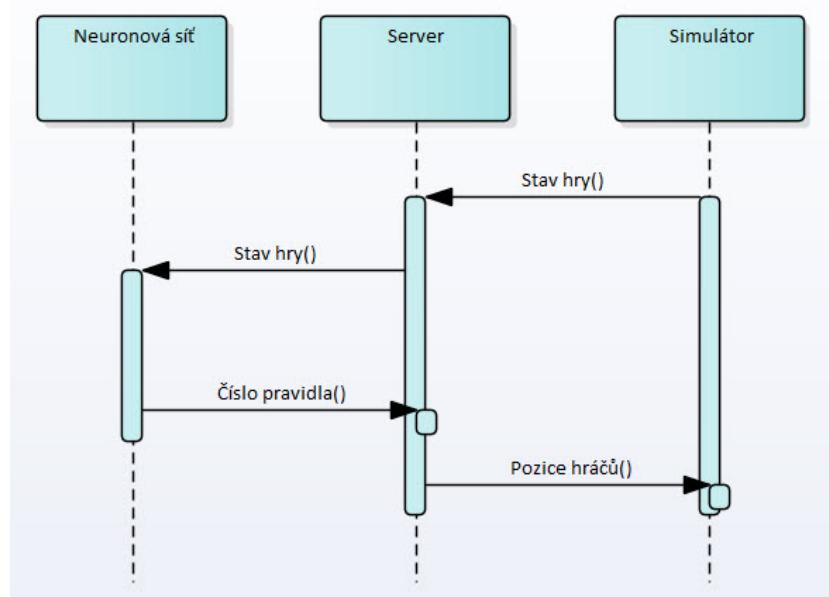


Obrázek 21: Grafy průměrné chyby na testovací množině

### 5.3.1 Komunikace simulátoru se sítí

Nyní již byly vytvořeny potřebné podklady pro tvorbu sítě. Zbývalo ovšem ještě vyřešit, jakým způsobem si budou síť a simulátor předávat informace.

Bylo třeba zajistit komunikaci z obou stran, tedy jak ze strany neuronové sítě, tak ze strany simulátoru. Za tímto účelem vznikl skript v Pythonu, který vytvoří TCP/IP server, na který se připojil simulátor. Simulátor následně komunikoval s programem pomocí JSON zpráv, kdy předával stav hry, a skript ho předával neuronové síti. Ta vrátila číslo pravidla, které je pro tuto situaci nejlepší. Díky číslu pravidla server zjistil instrukce Move. Ty byly znova předány do simulátoru pomocí JSON zpráv. Celá komunikace je znázorněna na Obrázku 22.



Obrázek 22: Komunikace simulátoru se sítí

## 5.4 Ověření funkčnosti sítě

Aby došlo k ověření, že komunikace funguje správně a síť opravdu prošla správným procesem učení a vyhodnocuje herní situaci, byl vytvořen jednoduchý test. Smyslem testu bylo pomocí strategie poslat roboty pravého týmu na jedno konkrétní herní pole. Neuronová síť měla na základě souřadnice robotů pravého týmu určit správné pravidlo, které pošle roboty levého týmu do jiného rohu hřiště.

### 5.4.1 Vznik testovací sítě

Nejdříve bylo pomocí Pythonu vygenerováno několik tisíc řádků herního logu. V těchto řádcích byly jen dvě možné pozice robotů pravého týmu. A to buď souřadnice (1,1), tedy levý dolní roh, nebo (1,6), která představuje pravý dolní roh. Pozice robotů levého týmu a míče byly vybrány náhodně. V těchto řádcích platí, že je - li pozice robotů pravého týmu (1,1), pak je jako pravidlo levého týmu v logu zapsána jednička. Pokud je pozice soupeřových robotů (1,6), pak je jako pravidlo zapsána dvojka. Tyto vygenerované logy byly předloženy síti k tréninku.

Pravidlo č. 1 znamenalo pohyb robotů pravého týmu na souřadnice (1,1) a pravidlo č. 2 znamenalo pohyb robotů na souřadnice (6,1).

### 5.4.2 Test sítě

V průběhu samotného testu byl nejdříve spuštěn server zajišťující komunikaci. Poté byla do simulátoru nahrána strategie pro pravý tým. V první situaci byla vybrána taková, která poslala roboty na souřadnice (1,1). Odpověď sítě mělo být pravidlo č. 1, které pošle roboty levého týmu na souřadnice (1,4).

Tato situace skutečně nastala jak nám dokládá Obrázek 23, což znamená, že první část testu proběhla úspěšně.



Obrázek 23: Situace 1

Druhá část byla obdobná s tím rozdílem, že roboti pravého týmu byli posláni na souřadnice (6,1). Jako odpověď sítě bylo tedy očekáváno pravidlo č. 2, které mělo poslat roboty levého týmu na souřadnice (2,3).

I tento druhý test proběhl úspěšně, jak je vidět na Obrázku 24.

Z tohoto testu plyne, že síť se naučila rozpoznávat herní situace a na základě toho vybrat příslušné pravidlo. Zároveň úspěšně probíhala komunikace mezi simulátorem a sítí.



Obrázek 24: Situace 2

## 5.5 Zápasy neuronové sítě

Vzhledem k ověření úspěšného použití neuronové sítě ve fotbale robotů bylo možno uskutečnit zápasy mezi sítí a ručně nadefinovanými strategiemi.

Pro tvorbu sítí byly využity logy z Noci vědců. Byly vytvořeny celkem tři sítě.

První síť se natrénovala na všech dostupných datech.

Trénovací data druhé sítě byly upraveny. Bylo vybráno vždy 100 řádků před vstřelením gólu, což jsou dvě vteřiny hry. Jedná se tedy o opravdu bezprostřední situace před vstřelením gólu. Tato úprava byla zvolena, jelikož vstřelení gólu je smyslem fotbalu.

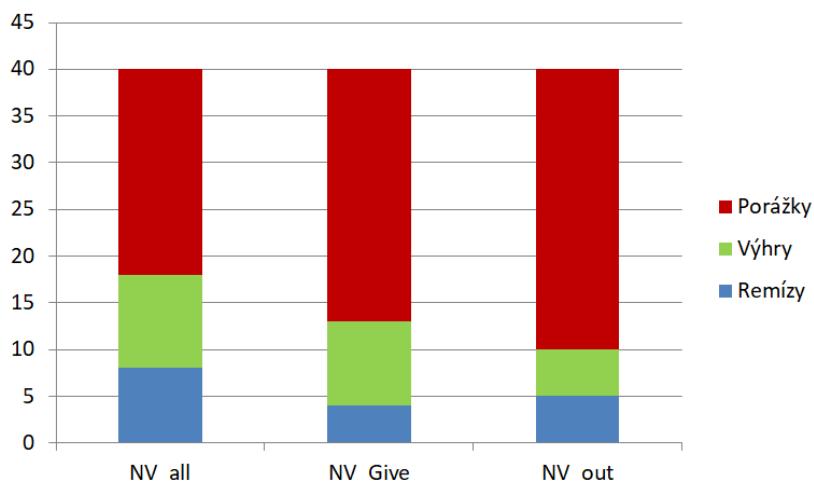
Pro třetí síť bylo naopak 100 řádků před obdržením nežádoucího gólu odebráno.

Zbývalo už jenom vybrat strategie, proti kterým bude síť hrát. Nabízelo se porovnat úspěšnost sítě proti ofenzivním a defenzivním strategiím, které byly použity již v předchozí části práce. Třetí strategií byla přesně ta, která byla připravena pro Noc vědců, kdy nebyla nějak upravována a ponechalo se všech 15 pravidel. Dále byla nově vytvořena čtvrtá strategie, která se skládala převážně z pravidel, která posílala roboty na různá, ne úplně logická místa, jako jsou například rohy herního pole.

### 5.5.1 Vyhodnocení zápasů

Každá ze sítí hrála vždy deset zápasů proti každé ze čtyř výše popsaných strategií viz 5.5.

Síť vytvořená ze všech dat nesla název *all*, síť natrénovaná na 100 řádcích před vstřelením gólu byla pojmenovaná *Give*, síť ze které bylo 100 řádků odebráno byla označena *Out*. Výsledky jednotlivých zápasů se lišily, jak je vidět na grafu na Obrázku 25. Síť natrénovaná na všech datech se ukázala jako nejúspěšnější. Ze všech sehraných zápasů dokázala 8 vyhrát, 10 remizovat a 22 prohrát. Zbylé dvě sítě nebyly tak úspěšné, a počet porážek byl vyšší.



Obrázek 25: Graf výsledků zápasů neuronových sítí z dat Noci vědců

Tyto výsledky svědčí o tom, že množství trénovacích dat je jedním z parametrů, které mají vliv na kvalitu naučení sítě. Během tréninku sítě je ovšem také důležité odstraňovat duplicity, tedy shodné herní situace, což bylo v rámci této práce splněno.

Abychom se podívali na průběh zápasu, byly opět vygenerovány teplotní mapy. Z těch bylo patrné, že strategie se drží logiky, ve které byly napsány. Mapa pro neuronovou síť naopak obsahuje kombinaci různých herních situací, jak ofenzivních tak defenzivních. Což souhlasí se vstupními daty, jelikož během Noci vědců byly vybrány různé typy pravidel. Cílem bylo vytvořit síť, která pokryje všechny herní situace.

### 5.5.2 Porovnání různých vstupních dat

Vzhledem k předchozím výsledkům zápasu byla vytvořena nová vstupní data, která čítala celkem 4 818 logů. Pro tvorbu strategií byla znova zvolena praví-

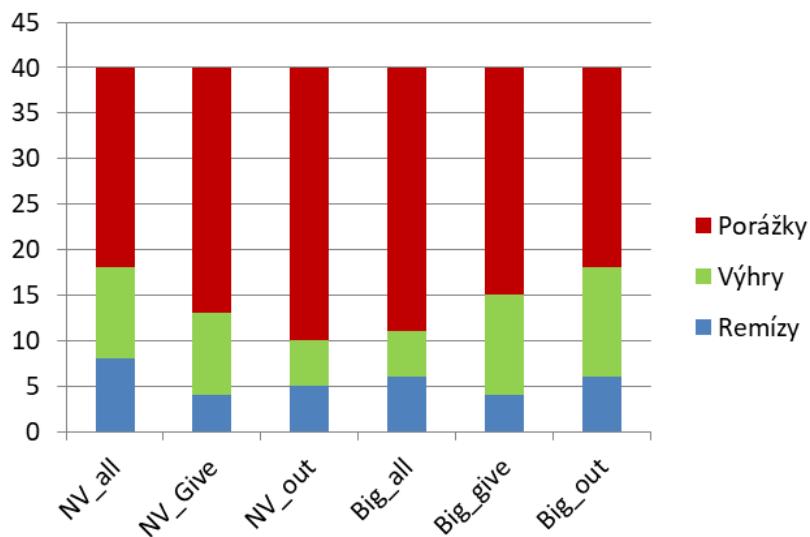
dla z Noci vědců. Jednotlivé strategie obsahovala vždy čtyři pravidla a jejich výběr pro každou strategii byl čistě náhodný. Sehráním zápasů těchto uměle vytvořených strategií vznikly požadované logy.

### 5.5.3 Vliv dat na výsledky zápasů

Znovu byly vytvořeny tři sítě se stejnou úpravou vstupních dat. Stejné byly i zvolené strategie, s kterými soupeřily.

Vzhledem k tomu, že pro jednotlivé sítě byly výsledky opět velmi rozdílné vznikl pro jejich názorné srovnání graf viz. Obrázek 26. Zelená barva reprezentuje počet vyhraných zápasů, modrá remízy a červená porážky. Sítě, k jejichž vytvoření byly využita data z Noci vědců nese označení NV. Sítě natrénovaná na náhodně vygenerovaných datech nese označení BIG. Označení úpravy dat je shodné s Sekcí 5.5.1. Jak můžeme vidět, přestože tři ze sítí byly natrénovány na daleko větší množině vstupních dat nebyly úspěšnější, pouze jedna z nich dosáhla stejně úspěšnosti jako ta, co byla natrénována na všech datech z Noci vědců. Z toho vyplývá, že důležitá je také reálnost dat. Uměle vytvořené strategie a logy vznikaly zcela na základě náhody, kdežto logy z Noci vědců byly vytvořeny na základě reálných her.

Pro výrazné zlepšení kvality sítě by ovšem bylo třeba získat daleko větší vzorek reálných dat.



Obrázek 26: Graf výsledků zápasů všech neuronových sítí

## 6 Závěr

Cílem této práce bylo přiblížit problematiku neuronových sítí a jejich využití v rámci her fotbalu robotů. Nejdříve bylo třeba nastudovat problematiku neuronových sítí, jejich princip a tvorbu.

Poté došlo k analýze zápasů sítě sehraných v simulátoru. Ta potvrdila fungování strategií a dodala potřebné informace, jako je popis herní situace, pro tvorbu neuronové sítě.

Následně bylo vytvořeno celkem šest neuronových sítí. Zároveň došlo k upravení samotného simulátoru, aby byla možná komunikace ze sítí. Toto dopadlo úspěšně a zápasy mezi simulátorem a neuronovou sítí se uskutečnily.

Vzniklé sítě se naučily aplikovat pravidla na příslušné herní situace. Ty ovšem pramenily pouze ze zaznamenaných situací v logu, díky čemuž se sítím nepodařilo získat převahu nad ručně nadefinovanými strategiemi.

Tato práce otevřela další možnosti využití fotbalu robotů a jeho propojení s umělou inteligencí. Pro budoucí zlepšení sítě by bylo vhodné použít pokročilejší metody učení, např. reinforcement learning. Ten byl použit pro již zmíněnou umělou inteligenci ve hře Dota II. Aby toto učení proběhlo úspěšně, muselo by dojít k nasimulování velkého počtu her. Na začátku by síť aplikovala pravidla náhodně, aby prozkoumala co nejvíce možností a postupně by vyhodnocovala, jaký vliv mají použitá pravidla na výsledek zápasu. Časem by se naučila preferovat ty kombinace pravidel, které vedou do vítězného konce.

Během celé tvorby práce jsem se ve velké míře zdokonalila v programování a získala spoustu motivace pro další výzvy, což osobně považuji za jeden z největších přínosů. Zároveň jsem měla možnost seznámit se s prostředím na IT4I a vyzkoušet si práci s týmem. Také to byla velmi cenná zkušenost.

## 7 Seznam referencí

### Odkazy

- [1] F. Rosenblatt. *The Perceptron, a Perceiving and Recognizing Automation Project Para.* Report: Cornell Aeronautical Laboratory. Cornell Aeronautical Laboratory, 1957. URL: [https://books.google.cz/books?id=P%5C\\_XGPgAACAAJ](https://books.google.cz/books?id=P%5C_XGPgAACAAJ).
- [2] Teuvo Kohonen. “Self-organized formation of topologically correct feature maps”. In: *Biological Cybernetics* 43.1 (led. 1982), s. 59–69. ISSN: 1432-0770. DOI: 10.1007/BF00337288. URL: <https://doi.org/10.1007/BF00337288>.
- [3] Yann LeCun et al. “A theoretical framework for back-propagation”. In: *Proceedings of the 1988 connectionist models summer school.* Sv. 1. CMU, Pittsburgh, Pa: Morgan Kaufmann. 1988, s. 21–28.

## 8 Seznam obrázků

### Seznam obrázků

1	Ukázka simulátoru . . . . .	8
2	Ukázka gridu . . . . .	8
3	Ukázka pravidla . . . . .	9
4	Ukázka StrategyCreatoru . . . . .	10
5	Ukázka herního logu . . . . .	11
6	Nákres biologického neuronu . . . . .	12
7	Model umělého neuronu . . . . .	12
8	Funkce sigmoid . . . . .	13
9	Struktura neuronové sítě . . . . .	13
10	Struktura práce . . . . .	15
11	Ukázka teplotní mapy . . . . .	16
12	Defenziva vs. Defenziva . . . . .	17
13	Ofenziva vs. Ofenziva . . . . .	18
14	Ofenziva vs. Defenziva . . . . .	18
15	Ukázka pohybu míče . . . . .	19
16	Ukázka z noci vědců . . . . .	19
17	Graf využití pravidel . . . . .	20
18	Znázornění herního plánu pomocí vícerozměrných polí . . . . .	21
19	Převod pozic objektů . . . . .	22
20	Grafy přesnosti sítě na trénovací množině . . . . .	23
21	Grafy průměrné chyby na testovací množině . . . . .	24
22	Komunikace simulátoru se sítí . . . . .	25
23	Situace 1 . . . . .	26
24	Situace 2 . . . . .	27
25	Graf výsledků zápasů neuronových sítí z dat Noci vědců . . . . .	28
26	Graf výsledků zápasů všech neuronových sítí . . . . .	29

## 9 Seznam příloh

Složka *Python-skripty* obsahuje

- Skript generující teplotní mapy pro míč *ball\_field.py*
- Hlavní modul pro tvorbu teplotních map *MainMenu.py*
- Skript generující teplotní mapy pro roboty *robot\_field.py*
- Skript vytvářející kombinace teplotních map *complete\_pict.py*
- Základní modul pro tvorbu neuronových sítí *menu-network.py*
- Skript pro vytvoření neuronové sítě *neural\_network.py*
- Skript zajišťující komunikaci mezi simulátorem a neuronovou sítí *server.py*
- Skript vytvářející trénovací data pro neuronovou síť *training\_data.py*

Složka *Ukázky* obsahuje

- Ukázku herního logu *ukazka\_log.log*
- Ukázku herní strategie *ukazka\_strategie.strg*
- Ukázku teplotní mapy *ukazka\_teplotnimapa.png*