POKYNY PRE RIEŠENIE A SPRACOVANIE ZÁVEREČNÝCH PRÁC

(spracované pre potreby študentov študijného programu ASI na Katedre informačných sietí Fakulty riadenia a informatiky Žilinskej univerzity v Žiline)

Pre spracovanie záverečných prác platí:

1. **Metodické usmernenie č. 56/2011 o náležitostiach záverečných prác, ich bibliografickej registrácii, uchovávaní** a sprístupňovaní (ďalej len Metodické usmernenie), vydanom Ministerstvom školstva, vedy, výskumu a športu SR, **platné od 1.9.2011. Usmernenie je dostupné na** <http://www.fri.uniza.sk/stranka/legislativa-doktorandskeho-studia> - smernice o záverečných prácach.
2. **V Y H L Á Š K A 233 Ministerstva školstva, vedy, výskumu a športu Slovenskej republiky z 1. júla 2011,** ktorou sa vykonávajú niektoré ustanovenia zákona č. 131/2002 Z. z. o vysokých školách a o zmene a doplnení niektorých zákonov v znení neskorších predpisov (ďalej len Vyhláška).

**Podľa Metodického odporúčania Čl. 3. Odporúčaná štruktúra záverečnej práce (ďalej ZP) odseky**

1. *Záverečná práca obsahuje úvodnú časť, hlavnú textovú časť a záverečnú časť; záverečná práca môže obsahovať aj prílohy.*
2. *Úvodná časť záverečnej práce obsahuje v uvedenom poradí tieto náležitosti:*
   1. *obal,*
   2. *titulný list,*
   3. *zadanie,*
   4. *poďakovanie, ak ho autor uviedol*
   5. *abstrakt v štátnom jazyku,*
   6. *abstrakt v cudzom jazyku určenom vysokou školou,*
   7. *obsah,*
   8. *zoznam ilustrácií a zoznam tabuliek,*
   9. *zoznam skratiek a značiek (najmä v technických a prírodovedných študijných odboroch),*
   10. *slovník, ak ho autor uviedol.*

**Podľa Vyhlášky § 1 Obal záverečnej práce, rigoróznej práce a habilitačnej práce**

(1) Na obale záverečnej práce sa uvádzajú tieto údaje:

a) názov vysokej školy,

b) názov fakulty, ak je autor zapísaný na štúdium študijného

programu uskutočňovaného na fakulte,

c) evidenčné číslo, ak bolo určené,

d) názov záverečnej práce a podnázov záverečnej práce,

ak sa použil,

e) označenie záverečnej práce

1. bakalárska práca,

2. diplomová práca alebo

3. dizertačná práca,

f) meno, priezvisko, akademické tituly a vedecko-pedagogické

tituly autora,

g) rok predloženia záverečnej práce.

(3) Vzor obalu záverečnej práce je uvedený v prílohe č. 1. Vyhlášky.

Vzor titulného listu záverečnej práce je podľa § 2 odseku (3) uvedený v prílohe č. 2 Vyhlášky.

Podľa Metodického usmernenia Čl. 5 **Odporúčaná formálna úprava záverečných prác odsek (1) „*Formálna úprava záverečných prác vychádza z noriem:***

*STN ISO 2145: 1997. Dokumentácia. Číslovanie oddielov a pododdielov písaných dokumentov.,*

*STN ISO 214: 1998, Dokumentácia. Abstrakty (referáty) pre publikácie a dokumentáciu.,*

*STN ISO 690: 1998. Dokumentácia – Bibliografické odkazy – Obsah, forma a štruktúra.,*

*STN 01 6910: 1999. Pravidlá písania a úpravy písomností.,*

*STN ISO 690-2: 2001, Informácie a dokumentácia. Bibliografické citácie. Časť 2: Elektronické dokumenty alebo ich časti."*

**odsek (3) „Odporúčaný typ písma** je Times New Roman, veľkosť 12 a je jednotný v celej práci. Odporúčané nastavenie strany - riadkovanie 1,5; okraje zľava 3,5 cm, sprava 2 cm, zhora a zdola 2,5 cm, orientácia na výšku, formát A4.

**odsek (4) Odporúčaný rozsah** bakalárskej práce je 30 až 40 strán (54 000 až 72 000 znakov

vrátane medzier), diplomovej práce 50 až 70 strán (90 000 až 126 000 znakov),

dizertačnej práce 80 až 120 strán (144 000 až 216 000 znakov) a habilitačnej práce do 150 strán.

**Vzor pre úvodnú časť ZP podľa Vyhlášky je na stranách 2-11.**

**ŽILINSKÁ UNIVERZITA V ŽILINE**

FAKULTA RIADENIA A INFORMATIKY

BAKALÁRSKA PRÁCA

Jakub Dzurovčin

**Algoritmus učenia posilňovaním v úlohe sprievodcu**

Vedúci práce: Ing. Marek Baláž

Registračné číslo: 1393/2020

Žilina, 2021

**ŽILINSKÁ UNIVERZITA V ŽILINE**

FAKULTA RIADENIA A INFORMATIKY

BAKALÁRSKA/DIPLOMOVÁ/dizertačná PRÁCA

ŠTUDIJNÝ ODBOR:

MENO PRIEZVISKO

**Názov práce**

Žilinská univerzita v Žiline

Fakulta riadenia a informatiky

Katedra informačných sietí

Žilina, 2016

****Čestné Vyhlásenie****

Podľa Vyhlášky § 5 Čestné vyhlásenie

(2) Ak je autor v právnom vzťahu s vysokou školou, najmä ako študent, uchádzač v rigoróznom konaní,

uchádzač v habilitačnom konaní alebo zamestnanec, vypĺňa čestné vyhlásenie v informačnom systéme vysokej školy. Takto vyplnené čestné vyhlásenie doručí autor prevádzkovateľovi registra prostredníctvom informačného systému vysokej školy.

(3) Ak autor nie je v právnom vzťahu s vysokou školou podľa odseku 2, doručuje čestné vyhlásenie prevádzkovateľovi registra v listinnej podobe.

Čestne prehlasujem, že som prácu vypracoval samostatne s využitím dostupnej literatúry a vlastných vedomostí. Všetky zdroje použité v bakalárskej/diplomovej/dizertačnej práci som uviedol v súlade s predpismi.

Súhlasím so zverejnením práce a jej výsledkov.

...........................................

V Žiline, dňa 7.11.2016 Meno Priezvisko

Poďakovanie

**Príklad poďakovania (nie je predpísané)**

„*Moje poďakovanie patrí vedúcemu bakalárskej práce Ing. Marekovi Balážovi za odbornú pomoc, pripomienky a usmerňovanie pri tvorbe práce.*“

ABSTRAKT V ŠTÁTNOM JAZYKU

Podľa Metodického usmernenia Čl. 3 Odporúčaná štruktúra záverečnej práce, odsek (4) „***Abstrakt obsahuje informáciu o cieľoch práce, jej stručnom obsahu a v závere abstraktu sa charakterizuje splnenie cieľa, výsledky a význam celej práce.*** *Súčasťou abstraktu je 3 - 5 kľúčových slov. Abstrakt sa píše súvisle ako jeden odsek a jeho rozsah je spravidla 100 až 500 slov.“*

*Pre písanie abstraktu platí* STN ISO 214: 1998, Dokumentácia. Abstrakty (referáty) pre publikácie a dokumentáciu.

**Príklad písania abstraktu**

„DANČÍK, Juraj: *Autentifikácia, autorizácia a účtovanie v SIP*. [Bakalárska práca] – Žilinská univerzita v Žiline, Fakulta riadenia a informatiky, Katedra informačných sietí. – Vedúci: Ing. Pavel Segeč, PhD. – stupeň odbornej kvalifikácie: Bakalár v odbore Informatika. Žilina: FRI ŽU v Žiline, 2013. – 55 s.

Cieľom predloženej bakalárskej práce je vybudovať SIP AAA riešenie s autentifikáciou pomocou protokolu RADIUS. Teoretická časť práce obsahuje možnosti využitia AAA v IP sieťach a v protokole SIP. Pri využití v IP sieťach sú popísané protokoly RADIUS a TACACS+. Pri využití AAA v protokole SIP je popísané účtovanie a autentifikačné a autorizačné modely. Bližšie oboznamuje s HTTP digest autentifikáciou a S/MIME autentifikáciou. Teoretická časť tiež popisuje nové metódy na distribúciu certifikátov medzi účastníkmi SIP siete. Praktická časť oboznamuje čitateľa s použitým softvérom, popisuje jeho inštaláciu a uvádza presný postup, ako nakonfigurovať SIP AAA riešenie.

**Kľúčové slová:** SIP AAA, RADIUS, Konfigurácia AAA, Autentifikácia, Autorizácia

ABSTRAKT V CUDZOM JAZYKU

Bibliografický abstrakt v cudzom jazyku.

Text abstraktu v svetovom jazyku je potrebný pre integráciu do medzinárodných informačných systémov (napr. The Network Digital Library of Theses and Disserta-tions). Ak nie je možné jazykovú verziu umiestniť na jednej strane so slovenským ab-straktom, je potrebné umiestniť ju na samostatnú stranu (cudzojazyčný abstrakt nemožno deliť a uvádzať na dvoch stranách).

**Key words**: Word, Phrase

Obsah

[ZOZNAM OBRÁZKOV 12](#_Toc68346981)

[ZOZNAM TABULIEK 13](#_Toc68346982)

[ZOZNAM SKRATIEK 14](#_Toc68346983)

[Úvod 15](#_Toc68346984)

[1 Súčasný stav riešenia problematiky 19](#_Toc68346985)

[1.1 Strojové učenie 19](#_Toc68346986)

[1.1.1 Učenie bez učiteľa 19](#_Toc68346987)

[1.1.2 Učenie s učiteľom 19](#_Toc68346988)

[1.1.3 Učenie posilňovaním 20](#_Toc68346989)

[1.2 Učenie posilňovaním 20](#_Toc68346990)

[1.2.1 Základný princíp 20](#_Toc68346991)

[1.2.2 Tabuľkové metódy 21](#_Toc68346992)

[1.2.3 Umelé neurónové siete 24](#_Toc68346993)

[1.2.4 Algoritmy využívajúce neurónové siete 26](#_Toc68346994)

[1.2.5 DQN 26](#_Toc68346995)

[1.2.6 Metódy založené na politike 28](#_Toc68346996)

[1.2.7 A2C 31](#_Toc68346997)

[1.2.8 A3C 32](#_Toc68346998)

[1.2.9 PPO 32](#_Toc68346999)

[1.3 Frameworky pre neurónové siete 32](#_Toc68347000)

[2 Ciele práce 34](#_Toc68347001)

[3 Metodika práce a metódy skúmania 36](#_Toc68347002)

[4 Výsledky Práce 38](#_Toc68347003)

[4.1 Riešenie problému 38](#_Toc68347004)

[4.2 Výsledky riešenia 38](#_Toc68347005)

[Záver 39](#_Toc68347006)

[Zoznam použitej literatúry 40](#_Toc68347007)

[Prílohy 42](#_Toc68347008)

[Príloha A: 43](#_Toc68347009)

[Príloha B: Obsah DVD 44](#_Toc68347010)

[5 Ďalšie usmernenia pre písanie záverečnej práce 45](#_Toc68347011)

[5.1 ČLENENIA KAPITOL A PRÁCE S ILUSTRÁCIAMI, TA-BUĽKAMI, ROVNICAMI A KRÍŽOVÝMI ODKAZMI 45](#_Toc68347012)

[5.1.1 Názov tretej úrovne 45](#_Toc68347013)

[5.2 Ilustrácie 46](#_Toc68347014)

[5.3 Tabuľky 48](#_Toc68347015)

[5.4 Rovnice, vzorce 48](#_Toc68347016)

[5.5 Krížové odkazy 49](#_Toc68347017)

[5.6 Skratky a odborné pojmy 49](#_Toc68347018)

[5.7 Odrážky 49](#_Toc68347019)

[5.8 Gramatika 49](#_Toc68347020)

[5.9 Obhajoby 49](#_Toc68347021)

# ZOZNAM OBRÁZKOV

[Obrázok 1: Účtovanie pri protokole RADIUS. 20](#_Toc433963509)

# ZOZNAM TABULIEK

[Tabuľka 1: Organizačné schémy a štruktúry IA (Morville – Rosenfeld, 2006) 32](#_Toc56679329)

# ZOZNAM SKRATIEK

|  |  |  |
| --- | --- | --- |
| ANSA | – | Automated Network Simulation and Analysis |
| BFD | – | Bidirectional Forwarding Detection |
| BGP | – | Border Gateway Protocol |
| Cisco IOS | – | Cisco Internetwork Operating System |
| CSPF | – | Constrained Shortest Path First |
| DLCI | – | Data Link Connection Identifier |
| DSCP | – | Differentiated Services Code Point |
| ECMP | – | Equal Cost Multiple Paths |
| FIR | – | Failure Insensitive Routing |
| GPL | – | General Public License |
| IDE | – | Integrated Development Environment |
| IGMP | – | Internet Group Management Protocol |

# Úvod

**„Moorov zákon** je empirické pravidlo, že zložitosť integrovaných obvodov (počet tranzistorov integrovaných na nich) sa zdvojnásobuje približne každé dva roky“[1]. Toto empirické pravidlo platilo posledných 5 dekád a jeho dôsledkom je obrovský nárast výkonu naších počítačov a aplikácií pre ktoré ich využívame.

V tejto práci priblížime strojové učenie, oblasť štúdie algoritmov ktoré sa dokážu učiť nové veci a zlepšovať sa vo vykonávaní svojej činnosti.

V teoretickej časti tejto práce popíšeme strojové učenie, jeho základné princípy, postupy ktorými sa dá dosiahnuť učenie. Bližšie sa pozrieme na učenie posilňovaním (Reinforcement learning) pomocou umelých neurónových sieti (Artificial neural networks), ich výhody, nevýhody a konkrétne algoritmy. Zbežne popíšeme druhy týchto algoritmov a podrobne vysvetlíme princíp fungovania Advantage Actor Critic (A2C) algortitmu. V závere teoretickej časti spomenieme novšie a výkonnejšie algoritmy ktoré su vylepšením A2C.

V praktickej časti rozanalyzujeme implementáciu A2C algoritmu a jeho vlastnosti. Popíšeme postup pri trénovaní takéhoto algoritmu a vytvoríme jednoduchú aplikácie na ktorej ukážeme možný príklad využitia natrénovaného algoritmu ako asistenta ktorý nám bude radiť pri hraní Pacmana.

Podľa Metodického usmernenia Čl. 3 Odporúčaná štruktúra záverečnej práce:

odsek (3)

„*Hlavnú textovú časť záverečnej práce tvorí:*

*a) úvod,*

*b) jadro,*

*c) záver,*

*d) resumé, ak je práca vypracovaná v inom ako štátnom jazyku,*

*e) zoznam použitej literatúry.“*

**odsek (6)**

*„V* ***úvode autor stručne a výstižne charakterizuje stav poznania*** *alebo praxe v oblasti, ktorá je predmetom záverečnej práce a oboznamuje čitateľa s významom, cieľmi a zámermi práce. Autor v úvode zdôrazňuje, prečo je práca dôležitá a prečo sa rozhodol spracovať danú tému.“*

odsek (7)

***„Jadro*** *je hlavná časť práce a jeho členenie je určené typom práce. Vo vedeckých a odborných prácach* ***má jadro spravidla tieto hlavné časti****:*

1. ***súčasný stav riešenej problematiky doma a v zahraničí,***
2. ***cieľ práce,***
3. ***metodika práce a metódy skúmania,***
4. ***výsledky práce,***
5. ***diskusia.“***

**Písanie odborného textu**

Pri písaní záverečných prác je vhodné dodržiavať nasledovné zásady písania odborného textu:

1. *V odbornom texte sa využívajú štylistické prostriedky, ktoré sa vyhýbajú opakovanému použitiu slova ja. Jedným je užívanie autorského alebo tiež skromného plurálu, ktorý je obvyklý v definičných procesoch (Za astroláb označujeme,..). Neosobné vyjadrovanie vyjadríme aj použitím pasívnych konštrukcií (Tretia kapitola je doplnená.., boli popísané,…).*
2. *Zo slovných spojení sa v odbornom texte objavujú spojenia, ktoré odkazujú:*
   1. *miestne: vyššie / nižšie uvádzame, v nasledujúcej kapitole,..*
   2. *časovo: doteraz sme pojednávali .., skôr než pristúpime k výkladu…*
   3. *procesuálne: ak sa vrátime k funkcii.., zakončili sme ..*
3. *Myšlienky vlastné sú oddeľované od myšlienok prevzatých, ktoré citujeme.*

**Bibliografické citácie**

Bibliografické citácie upravujú normy:

* STN ISO 690 Informácie a dokumentácia. Návod na tvorbu bibliografických odkazov na informačné pramene a ich citovanie, 2012.
* STN EN ISO 690-2 Informácie a dokumentácia - Bibliografické citácie. Časť 2. Elektronické dokumenty alebo ich časti, 2001.

Základné pojmy citovania podľa normy STN ISO 690 sú:

* **Bibliografický odkaz** (angl. references) je informácia o dokumente, ktorý sa skutočne použil pri písaní práce. Je súčasťou zoznamu použitej literatúry.
* **Citácia,** (angl. citacion) je skrátený odkaz umiestnený v zátvorkách vnútri textu alebo pripojený k textu ako poznámka (v dolnej časti strany, na konci kapitoly alebo na konci textu). Umožňuje presne identifikovať publikáciu z ktorej sa preberala myšlienka alebo údaj a označuje jej presné umiestnenie. **Citácia sa spravidla viaže na bibliografický odkaz**.
* **Bibliografia** je usporiadaný zoznam záznamov o literatúre, ktorá sa týka predmetu práce, ale ktorá nebola bezprostredne použitá na jej napísanie. Môže sa pridať k práci ako samostatná časť.
* **Priama citácia** je doslovný text vložený do práce iného autora – ten je treba vždy dať do úvodzoviek a citovať.

Nové vydanie normy ISO 690 (2010) odporúča tri techniky/metódy citovania:

**1**. Techniku 1. údaja a dátumu (harvardský systém),

2. Techniku číselných odkazov (numerický systém),

3. Techniku priebežných poznámok.

Všetky tri techniky sú rovnocenné, avšak platí zásada používania len jednej techniky v celom citujúcom dokumente. **V prípade,  že má vedná disciplína alebo vydavateľ vyšpecifikovanú vlastnú metódu citovania, nie je nutné dodržiavanie techník ISO 690  (2010).**

dokumentu.

V záverečných prácach odporúčame Techniku číselných odkazov v hranatých zátvorkách, príklad: [2].

Všetky citácie toho istého dokumentu majú **rovnaké poradové číslo** ako prvá citácia tohto diela.

**Bibliografické odkazy** sú v zozname literatúry usporiadané podľa poradových čísel; odkazy sú odsadené. Príklady písania bibliografických odkazov sú v časti Zoznam použitej literatúry.

**Kapitoly prvej úrovne začínajú vždy na novej strane.** Na oddeľovanie týchto strán používajte vkladanie zlomov strán (Vložiť -> Zlom strany).

# Súčasný stav riešenia problematiky

## Strojové učenie

Strojové učenie je zastrešujúci pojem pre rodinu algoritmov ktoré sú schopné plniť svoju danú úlohu bez toho aby mali explicitne naprogramovaný postup pre jej plnenie.

Pre dosiahnutie tohto cieľa oblasť strojového učenia popisuje metódy a algoritmy ktoré sa dokážu učiť z poskytnutých dát a zlepšovať svoju presnosť pri plnení danej úlohy.

V čase písania tejto práce existujú 3 základné metódy ako pristupovať k učeniu.

### Učenie bez učiteľa

Pri učení bez učiteľa su použíté neoznačené dáta. Algoritmus nevie čo tieto dáta predstavujú. Podľa aplikácie jeho úlohou môže byť:

* Hľadanie spojitosti a štruktúry ktorá sa v daných dátach nachádza.
* Identifikácia výnimiek skrývajúcich sa v daných dátach.

Algoritmy využivajúce túto metódu sa použivajú napríklad v marketingu kde dokážu identifikovať zákazníkov s podobnými atribútmi.

### Učenie s učiteľom

Učenie s učiteľom využíva označené dáta, kde algoritmus dostane konkretný údaj aj s „popisom“ čo daný údaj predstavuje. Algoritmus spracúva tieto dáta a predikuje čo tieto dáta predstavujú. Po vytvorení predikcie svoj odhad porovná s nálepkou a upraví svoj vnútorný model. Po natrenovaní dokáže tento algoritmus robiť predikcie aj pre neoznačené dáta. Učenie s učiteľom sa používa v aplikáciach kde sa na základe historických dát dajú predpovedať budúce udalosti.

### Učenie posilňovaním

Pri učení posilňovaním je algoritmu poskytnuté prostredie v ktorom dokáže vykonávať akcie. V takomto prostredí algoritmus pomocou metódy pokus omyl vykonáva akcie za ktoré je odmenený a jeho cieľom je maximalizovať tieto odmeny.

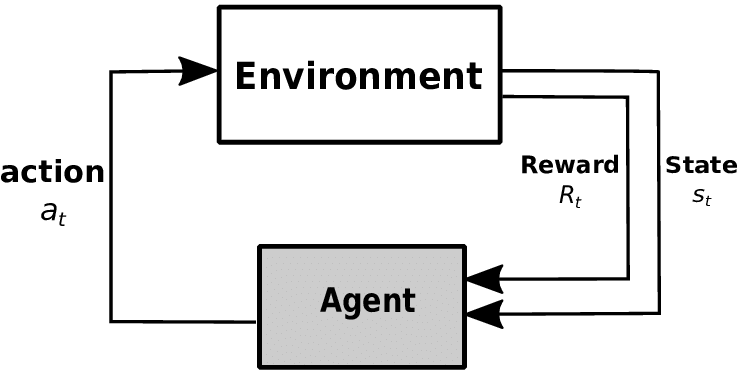
## Učenie posilňovaním

### Základný princíp

Pri učení posilňovaním je našim cieľom vytvoriť algoritmus - agenta ktorý vie vybrať najlepšiu možnú akciu z množiny všetkých možných akcií v danom prostredí.

Pre dosiahnutie tohto cieľa musí mať agent prístup k prostrediu (Environment) v ktorom može vykonávať kroky. Krok zvyčajne pozostáva z akcie () vybranej agentom ktorá po vstupe do prostredia vyprodukuje stav () čo je aktuálny stav nášho prostredia a odmena () za vykonanie tejto akcie.

Počítačové hry majú častokrát systém v ktorom hráč vykonáva akcie nad herným prostredím a úspešnými akciami zvyšuje svoje skóre. Preto sú hry ukážkovým prostredím pre vývin a testovanie agentov využívajúcich tento princíp učenia.



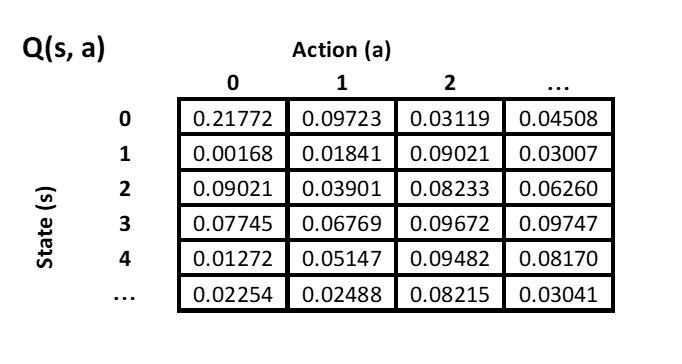
Obrázok 1.1: Princíp učenia posilňovaním

### Tabuľkové metódy

Existuje viac prístupov k učeniu posilňovaním. Jedným z nich sú tabuľkové metódy. Základom tabuľkových metód je takzvaný Q - learning algoritmus ktorý na učenie využíva Q value.

**Q – learning algoritmus**

Hlavným komponentom Q – learning algoritmu je Q tabuľka, čo je dvojrozmerné pole tvaru ktoré sa na začiatku inicializuje – zvyčanie na 0.



Obrázok 1.2: Vizualizácia Q – tabuľky

Agent začína v stave () kde vykoná akciu () za ktorú dostane odmenu ().

Vyberanie akcie môže nastať dvojako: využitím (Q tabuľky) alebo objavovaním.

Pri objavovaní sa zvolí náhodná akcia pre daný stav. Týmto spôsobom dokáže agent objaviť nové stratégie ktoré nie su zarátané v Q tabuľke.

Pri využívaní Q tabuľky agent vyberá akciu s najvyššou hodnotou (Q value) pre daný stav v ktorom sa nachádza.

Pomer medzi objavovaním a učením sa dá nastaviť pomocou „Epsilon Greedy“ metódy. V jej najjednoduchšej podobe môžme pomer medzi objavovaním a využívaním určiť fixne: napríklad 1 z 10 akcii bude náhodná.

Po vykonaní akcie sa aktualizujú hodnoty Q tabuľky. Na tento účel slúži vzorec pre počítanie Q hodnoty:

Rovnicu Q algoritmu môžme rozdeliť na niekoľko častí:

Táto časť sa nazýva aj dočasný rozdiel (TD – temporal difference)

Najvyššia Q hodnota ktorú je možné získať v nasledujúcom stave (Vybraním najhodnotnejšej akcie tohto stavu) znížená o  Q hodnotu aktuálneho stavu a akcie.

(diskontný factor, gamma) používame na zníženie dôležitosti budúcich Q hodnôt. Pokiaľ je gamma bližšie k 0 tak rozdiel medzi budúcou a aktuálnou Q hodnotou zaváži menej. Pri gamma blížiacej sa k 1 bude rozdiel budúcej vs okamžitej hodnoty viac ovplyvňovať aktuálnu odmenu (zvýši ju) a tým pádom bude agent preferovať budúce odmeny.

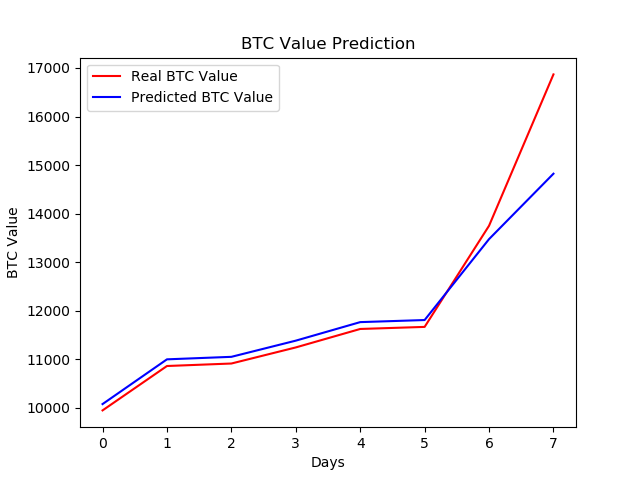
(miera učenia, learning rate, lr, alpha) predstavuje ako veľmi akceptujeme novú hodnotu (najlepšiu z následujúceho stavu) oproti aktuálnej. Vo vzorci vyššie môžme vidieť: Mierou učenia vynásobime dočasný rozdiel (TD). Čím vyššia je naša miera učenia, tým vačší bude rozdiel medzi aktuálnymi a novými hodnotami - to má za následok rýchlejšie zvyšovanie Q hodnôt pri aktualizáciach.

**Aktuálny stav**

Kvôli potrebe zostrojenia tabuľky sú tieto metódy efektívne v prostrediach s menším počtom stavov a akcií. Pre zložitejšie aplikácie kde zostrojenie tabuľky neprichádza do úvahy sa používajú metódy využívajúce hlboké neurónové siete.

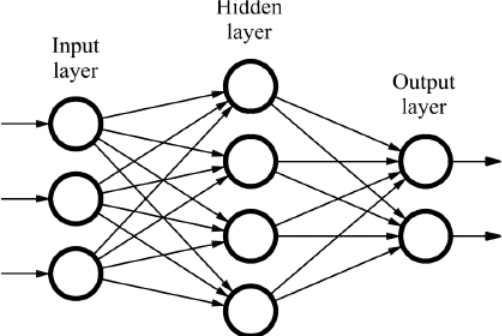
### Umelé neurónové siete

Umelé neurónové siete sú počítačový systém určený na predikciu výstupných hodnôt z poskytnutých vstupných hodnôt. Boli vyvynuté po vzore neurónových sieti v ľudskom mozgu a do istej miery kopírujú ich funkcionalitu.



Obrázok 1.3: Príklad použitia neurónovej siete na predikciu ceny bitcoinu

Umelé neurónové siete sa skladajú z uzlov – umelých neurónov ktoré sú usporiadané po vrstvách. Každá sieť obsahuje vstupnú (input layer) a výstupnú (output layer) vrstvu a každý umelý neurón vo vstupnej vrstve je prepojený s každým umelým neurónom vo výstupnej vrstve. Pridaním ďalších vrstiev medzi vstupnú a výstupnú vrstvu – nazývaných skryté vrstvy (hidden layer) a ich následným prepojením získame neurónovú sieť ktorú nazývame hlboká neurónová sieť.

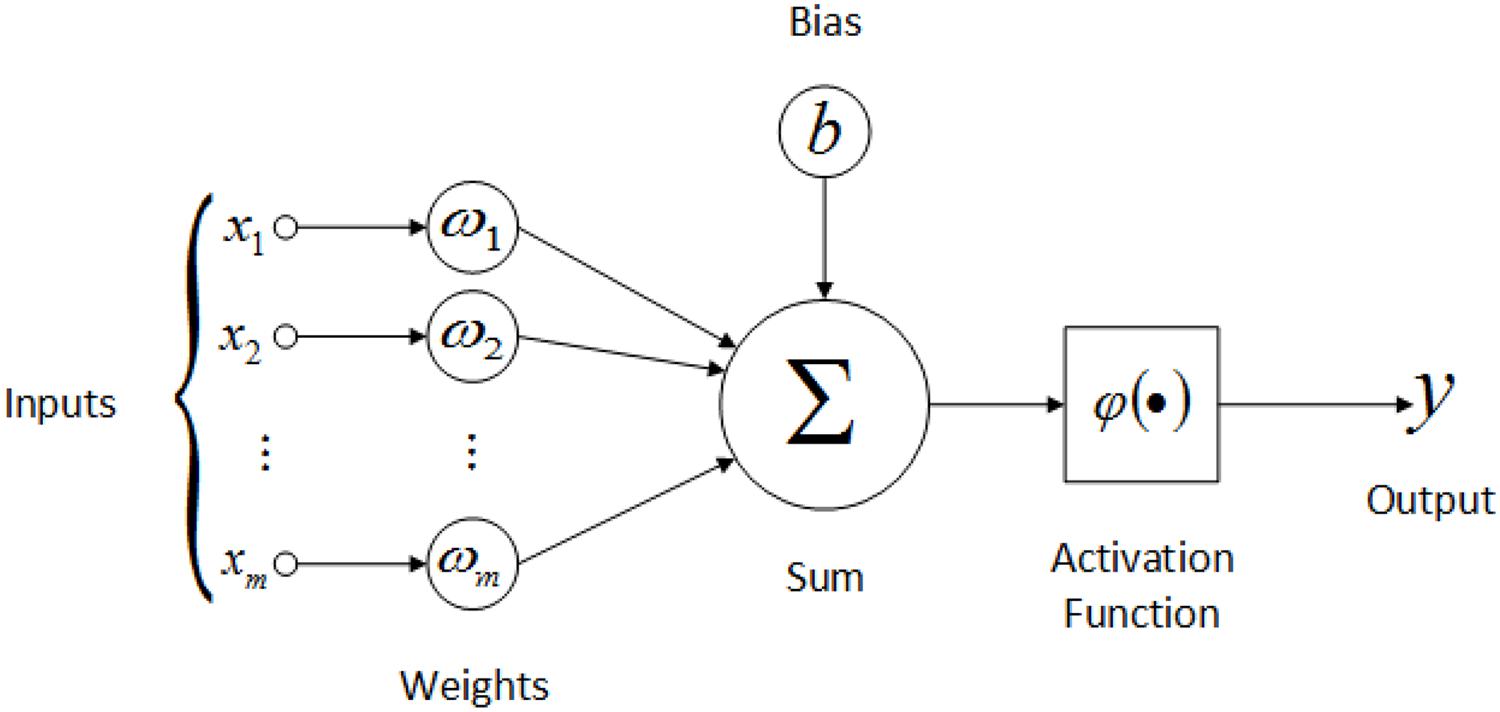


Obrázok 1.4: Hlboká neurónová sieť

Vstupné dáta prechádzajú touto sieťou vrstvu po vrstve od vstupnej po výstupnú. Toto sa nazýva dopredná propagácia. Každý umelý neurón dostane pole vstupov z predchádzajúcej vrstvy ktoré vynásobi svojími váhami (weights - ). Neurón obsahuje jednu váhu pre každy vstup . Po vynásobení sa pripočíta bias - *b* a spustí sa aktivačná funkcia . Výstup (*y*) po aplikovaní aktivačnej funkcie pokračuje do neurónov v ďalšej vrstve až kým sa nedostaneme na konečný výstup z neurónovej siete.

Váhy a bias sú naučitelné parametre ktoré sa nastavujú v procese ktorý nazývame trénovanie neuróvej siete.

Aktivačná funkcia je nelineárna funkcia vďaka ktorej vie neurónová sieť aproximovať komplexné funkcie.



Obrázok 1.5: Umelý neurón

Pri trénovaní neurónovej siete sa určí funkcia ktorá definuje cieľ ktorý chceme dosiahnúť. Môže byť ohodnocovacia (chceme čo najvyššie ohodnotenie - maximalizujeme) alebo stratová (chceme znížť stratu - minimalizujeme) . Po dosadení príslušných hodnôt do funkcie a vypočítaní straty sa táto hodnota spätne propaguje do neurónovej siete a nastavujú sa podľa nej váhy a bias.

### Algoritmy využívajúce neurónové siete

Pri použití neurónových sieti pre učenie posilňovaním rozlišujeme dva základné prístupy:

**Hodnotovo založené metódy** – vychádzajú z Q – learning algoritmu. Pri tomto prístupe vymeníme Q tabuľku za Q funkciu. Neurónová sieť potom slúži ako aproximátor Q funkcie. Základný algoritmus založený na tomto princípe sa nazýva DQN (Deep Q learning)

**Metódy založené na politike** – Pri tomto prístupe neurónová sieť priamo aproximuje politku (funkciu) pre vyberanie akcií.

Zkombinovaním týchto prístupov dostávame Actor-critic metódy ktoré využívajú oba prístupy.

### DQN

Vymenením Q tabuľky za hlbokú neurónovú sieť sa mení spôsob aktualizácie Q hodnôt. Teraz namiesto priamého počítania novej Q hodnoty a vkladania ju do tabuľky počítame stratu (loss) ktorá vznikla pri predikcii Q hodnoty.

Podobne ako pri Q learning algoritme - strata sa počíta ako rozdiel najväčšej možnej Q hodnoty v budúcom stave a Q hodnoty predikovanej pre aktuálny stav.

Pre stabilizovanie učenia sa pri DQN používajú dve rovnaké neurónové siete - cieľová sieť (target) a prediktívna sieť (prediction). Cieľova sieť je zmrazená na N krokov a strata je počas týchto krokov spätne propagovaná len cez prediktívnu sieť. Inými slovami: Aktualizujú sa len parametre patriace prediktívnej sieti Po N krokoch sa vypočíta sa nová strata a aktualizuje sa cieľová sieť podľa prediktívnej siete. Následne prediktívna sieť pokračuje v učení podľa novej vypočítanej straty.

Predstavuje parametre siete ktoré sa pri trénovaní nastavujú. Vľavo sú parametre cieľovej siete a vpravo parametre prediktívnej siete

Ďalším trikom ako zlepšiť učenie pri použití neurónových sieti je použitie pamäte (memory replay). Po sebe nasledujúce stavy sú korelované. Preto vytvoríme pamäť do ktorej budeme ukladať naše stavy a pri trénovaní budeme mixovať stavy z prostredia so stavmi ktoré máme zapamätané.

**Ďalšie algoritmy odvodené z DQN**

**Double DQN –** DQN algoritmus používa tú istú neurónovú sieť pre ohodnotenie akcie Q hodnotou a jej vybratie. Dôsledok toho je občasné „preháňanie“ (overestimation) pri výbere akcii[3]. Double DQN rieši tento problém použitím dvoch sieti kde: Jedna je použitá na vyberanie akcií a druhá na generovanie Q hodnoty.

**Duelling DQN** – Duelling DQN počíta Q hodnotu pomocou dvoch konceptov:

(advantage - *A*) nám vraví o koľko je vyberaná akcia lepšia ako ostatné akcie.

(value - *V*) predstavuje hodnotu stavu v ktorom sa nachádzame.

Pri duelling prístupe využívame dve siete. Jednu pre aproximáciu a druhú pre aproximáciu kde na konci ich hodnoty spojíme dokopy. Kvôli povahe implementácie ale spájanie nemôže byť obyčajný súčet je nutné využiť jeden z dvoch trikov popísaných v originálnej publikácií [4], kde nájdete aj plnú špecifikáciu tejto architektúry.

### Metódy založené na politike

Metódy založené na politke majú niekoľko výhod oproti hodnotovo založeným metódam:

Metódy založené na politike majú lepšiu konvergenciu (nájdenie optimálneho stavu). Metódy vyberajú priamo akciu a preto pri menšej zmene politky nastane menšia zmena pri vyberaní akcií. Pri hodnotovo založených metódach môže mať malá zmena Q hodnoty drastický efekt na vybranú akciu (napríklad keď majú dve veľmi rozdielne akcie podobné Q hodnoty).

Politka môže byť deterministická alebo stochastická (hodnotovo založené metódy sú len deterministické). To znamená že naša akcia môže byť konkrétna hodnota, alebo rozdelenie pravdepodobností pre dané akcie. Stochastická politika je výhodná v prostrediach ktoré obsahujú elementy neistoty, kde je viac ako jedna správna akcia ktorú môžeme vykonať.

Stochastickým politikám sa darí lepšie v priestoroch s veľkým počtom akcií (hodnotovo založené metódy musia ohodnotiť každú akciu pre daný stav) alebo v prípadoch kde je nutné použiť spojité akcie (Napríklad regulovať plynový pedál). Tak isto stochastická politka odstraňuje problém objavovania vs využívania (funkcie). V prípadoch keď majú dve akciu rovnakú pravdepodobnosť môžeme náhodne vybrať jednu z nich, čiže objavovanie je zabudované v princípe tejto metódy a nemusí sa pridávať externe.

Nevýhody metód založených na politke:

Kvôli mechanizmu ktorý používajú pre optimalizáciu sú tieto metódy pomalšie na natrénovanie a vzorkovo menej efektívne ako hodnotovo založené metódy.

**Politka**

Politiku označujeme gréckym písmenom s parametrami . Vstupom do politky je akčný priestor a stav prostedia. Výstupom politky je rozdelenie pravdepodobnosti vykonania akcií *a* v stave *s*v prípade stochastickej politky*.*

Naším cieľom je optimalizovať parametre . Na to sa používa ohodnocovacia funkcia politiky (Policy Score function) **. T**áto funkcia hodnotí očakávanú odmenu ktorú môžeme dosiahnuť podľa danej politky . Existuje viacero implementácií tejto hodnotiacej funkcie:

V epizodických prostrediach kde je definovaný začiatok a koniec epizódy môžeme brať epizódu ako celok a bude predstavovat kumulovanú zdiskontovanú odmenu za každý krok v epizóde (napríklad najvyššie dosiahnuté skóre na konci hry).

V spojitých prostrediach ktoré nemajú začiatok ani koniec sa nemôžeme spoliehať na konkrétne stavy. Preto sa stavy spriemerovávajú a hodnotia sa podľa ich výskytu.

**Kde**

je počet výskytov aktuálneho stavu aje počet výskytov všetkých stavov.

Pokiaľ chceme dosiahnúť najvyššiu odmenu za určitý časový úsek tak použijeme tretí spôsob.

pravdepodobnosť že som v stave ***s.***

pravdepodobnosť vybratia akcie ***a***  v stave ***s***  pri nasledovaní politky **.**

Okamžitá odmena.

Samotná optimalizácia sa vykonáva pomocou gradientov. Gradienty sa počítajú z hodnotiacej alebo stratovej funkcie a určuje zmenu pre naše parametre . V tomto prípade máme hodnotiacu funkciu ktorú chceme maximalizovať, na to sa používa metóda stúpajúceho gradientu (Gradient ascent). V prípade že máme stratu ktorú chceme minimalizovať sa používa metóda klesajúceho gradientu (Gradient descent). Gradient označujeme ako .

V prípade že používame stochastickú politku je naším výstupom rozdelenie pravdepodobnosti pre akcie. Pre výpočet gradientu je ale nutné použiť derivácie nad týmto rozdelením, na čo sa používajú logaritmy[5].

Finálna podoba gradientu ohodnoteného pomocou :

Zmenu parametrov môžeme zapísať ako teda:

Konečným cieľom je nájsť nájsť optimálne parametre .

### A2C

A2C (Advantage Actor Critic) patrí do rodiny Actor-Critic algoritmov ktoré spájajú výhody metód založených na politike s výhodami metód založených na hodnotách. Využíva dve neurónové siete:

Actor – vyberá akcie podľa svojej politiky.

Critic – ohodnocuje akcie vybrané Actorom pomocou A hodnoty (advantage - o koľko je vybraná akcia lepšia ako ostatné akcie) ktorá sa následne použije pri učení lepšej politky pre Actora.

Algoritmy založené len na politke ako napríklad algoritmus REINFORCE vykonávajú aktualizácie až na konci epizódy. To znamená že sa učí zo všetkých krokov vykonaných počas epizódy, aj tých menej optimálnych, alebo vyslovene zlých. Kvôli tomu potrebujeme pri týchto algoritmoch viac vzoriek než pri algoritmoch založených na hodnote. Pridaním A hodnoty (advantage) dokážeme aktualizovať naše parametre kazdých ***t*** krokov algoritmu. Pre A hodnoty potrebujeme ďalšiu neurónovú sieť s vlastnou V funkciou (aproximuje hodnotu stavu) a vlastnými parametrami ***v***.

Actor – politika pre vyberanie akcií:

Critic – funkcia pre odhad V hodnoty:

**potom:**

Čo nám dá výsledný vzorec pre aktualizáciu parametrov:

A2C nemá pamäť (memory replay) ako napríklad DQN, preto sa pre odstránenie korelácií používa paralelné trénovanie, kde sa neurónová siet pridelí skupine pracovníkov (workerov) kde každý dostane svoje vlastné prostredie v ktorom každý vykonáva kroky nezávisle na ostatných pracovníkoch. Každých ***t*** krokov sa vykonané akcie, stavy a odmeny spriemerujú a následne sa vykoná aktualizácia a každý pracovník dostane aktualizované parametre.

### A3C

A3C (Asynchronous Advantage Actor Critic) je predchodcom A2C. Pri tejto architektúre používajú pracovníci kópiu globálnej neurónovej siete pomocou ktorej robia kroky vo svojích prostrediach. Keď pracovník vykoná určitý počet krokov, nezávisle na ostatných pracovníkoch tak aktualizuje globálnu sieť pomocou svojích parametrov a skopíruje aktuálny stav aby získal parametre ktoré aktualizovali ostatní pracovníci. Vďaka asynchrónnym aktualizáciam môžu jednotliví pracovníci pracovať vo vlastných vláknach a tým pádom byť efektívnejší čo sa využívania CPU týka. Zistilo sa ale že asynchrónnosť neposkytuje žiadne extra výhody, práve naopak. Podľa OpenAI Baselines synchrónna implementácia a využitie GPU vykazuje lepšie výsledky [6].

### PPO

PPO (Proximal Policy Optimization) bol vytvorený v roku 2017 tímom spoločnosti Open AI a v čase písania tejto práce je považovaný za jeden z najlepšich (state of the art) algoritmov pre učenie posilňovaním [7]. Existujú aj efektívnejšie algoritmy ako napríklad ACER, ktorý je ale podstatne zložitejší na implementáciu.

PPO patrí do rodiny Actor-Critic algoritmov ktoré zlepšuje úpravou Actor časti. A2C je citlivý na hyperparametre. Pri neoptimálnom nastavení môžeme po čase pozorovať stratu schopnosti učiť sa (príklad tohto javu sa nachádza aj v implementačnej časti tejto práce). Tento jav je spôsobený príliž veľkými zmenami pri optimalizácií parametrov. PPO rieši tento problém – vylepšuje stabilitu učenia a znižuje citlivosť na hyperparametre. Na dosiahnutie tohto cieľa vypočítame rozdiel medzi pravdepodobnosťami vykonania akcie pod novou a starou politikou.

Potom:

Akcia je preferovanejšia pod novou politikou,

Akcia je preferovanejšia pod starou politikou

Vďaka vieme povedať aká veľká zmena v politike nastane. Ďalši krok je ohraničenie veľkosti tejto zmeny. Podľa originálnej publikácie by mala byť táto zmena obmädzená v rozmädzí od 0,8 po 1,2 [8]. Na obmädzenie môžeme použiť jeden z dvoch spôsobov popísaných v publikácií. TRPO (Trust Region Policy Optimization) alebo tzv. Pristrihnutie (Clipping).

Pristrihnutie:

Takto dostaneme dva pomery pravdepodobnosti – nepristrihnutý a pristrihnutý pomer v rozsahu , kdeje hyperparameter ktorý predstavuje rozsah pristrihnutia (podľa publikácie ). Následne vezmeme minimum – čiže konzervatívnejšiu hodnotu pre zmenu politiky.

## Frameworky pre neurónové siete

V tejto časti predstavíme dva frameworky pre efektívne písanie algoritmov strojového učenia pomocou neurónových sieti.

TensorFlow – najpopulárnejší framework v čase písania tejto práce [9].

Pytorch – framework v ktorom sme implementovali náš algoritmus.

### TensorFlow

[9][10] Vyvynutý Google Brain tímom a napísaný v programovacích jazykoch: Python, C++ a CUDA. TensorFlow beží na všetkých populárnych operačných systémoch (Linux, Windows, MacOS, iOS, Android). Obsahuje podporu pre využitie CPU, GPU alebo TPU.

Funkcie poskytované TensorFlow frameworkom:

Data Flow Graphs – v TensorFlow sú výpočty popísané grafmi prúdenia dát. Uzly v grafe reprezentujú matematické operácie a hrany predstavujú viacrozmerné datasety (tensory). Tento prístup je výhodny v oblasti neurónových sietí kde sa dáta často menia a vykonáva sa nad nimi množstvo operácií. Grafy samotné sa dajú vizualizovať pomocou funkcie nazývanej TensorBoard pre ešte lepšie pochopenie toku dát a vykonaných operacií.

Knižnica funkcií – TensorFlow obsahuje širokú škálu už implementovaných matematických funkcií pripravených na použitie.

Podpora pre programovacie jazyky – Python (so stabilnými aktualizáciami), C++, Javascript, Java, Go, Haskell, C#, R, Julia, Scala, Ruby, MATLAB (vo vývoji, nestabilné aktualizácie).

### Pytorch

[9][11] Torch je počítačový framework orientovaný na vedecké aplikácie. Pytorch je python implementácia tohto frameworku. Považuje sa za kompetitora TensorFlowu a zdieľa s ním množstvo výhod:

Podpora pre všetky populárne operačné systémy,

Programovanie v Pythone s dostupným API pre C++

Silná podpora pre GPU výpočty.

Synergia s populárnymi Python knižnicami ako Numpy, SciPy, Pandas ...

Ďalšie vlastnosti:

TorchScript - metóda pre jednoduché písanie machine learning modelov,

TorchServe – umožnuje jednoduché nasadzovanie modelov do produkčných prostredí,

Paralelné výpočty,

Podpora ONNX štandardu,

Podpora pre cloud,

A mnohé ďalšie.

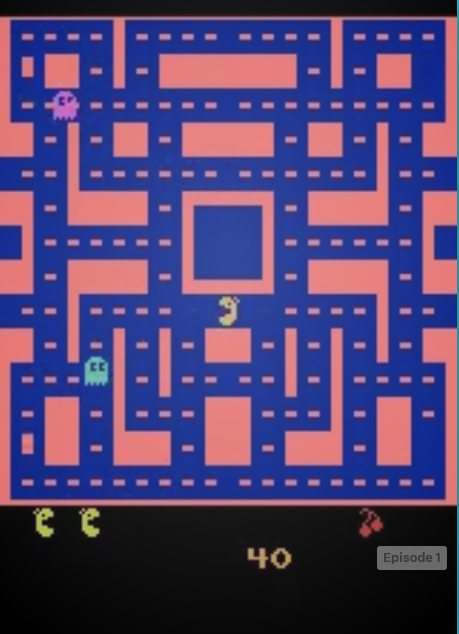
# algoritmus a2c

V tejto časti si popíšeme našu implementáciu A2C algoritmu. Celá implementácia je realizovaná v jazyku Python s pomocou frameworku PyTorch.

## Prostredie

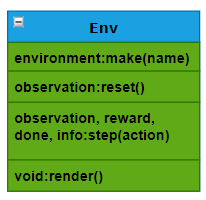
Pre vytvorenie našeho prostredia používame knižnicu „Gym“ od OpenAI. Táto knižnica ponúka širokú škálu prostredí založených na Atari hrách ktoré sú určené špecificky pre trénovanie neurónových sietí.

V tejto práci budeme používať prostredie Pacman (Ms-Pacman-v0).



Obrázok 2.1: Pôvodna obzervácia

Prostredie sa vytvára pomocou metódy **gym.make(‘Ms-Pacman-v0’)** a inicializuje pomocou **gym.reset()** čo nám zresetuje prostredie do východiskového stavu a vráti prvú obzerváciu (stav prostredia). Následne sa v ňom môžeme vykonávať kroky pomocou metódy **gym.step(akcia)**. Návratová hodnota tejto metódy je obzervácia po vykonaní akcie, odmena za vykonanú akciu, boolean ktorý indikuje či nastal koniec hry a diagnostické informácie pre potreby debbugingu. Vizualizácia prostredia je možná pomocou metódy **gym.render().**



Obrázok 2.2: UML prostredia

Pre efektívnejšie učenie sme použili obaľovače (Wrapper) pomocou ktorých sme upravili naše prostredie:

SkipEnv(4) – Preskočí 3 obzervácie, čiže dostaneme každý 4. stav a sumu odmien za dané stavy. Pacman používa malé kroky, preto je výhodné použiť tento wrapper.

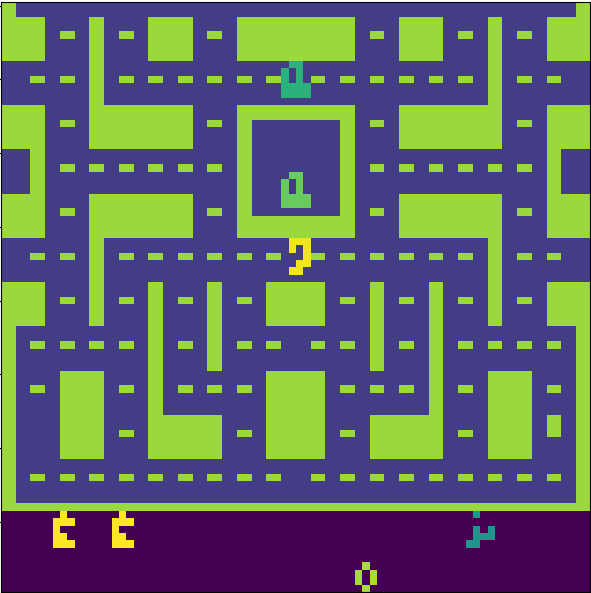
PreProccessFrame – Zmenší našu obzerváciu z veľkosti 210x160x3 na veľkosť 80x80 a odstraní RGB. Výsledná obzervácia je 80x80x1

MoveImgChannel – Zmení poradie prvkov v tensori. Originálna obzervácia je 80x80x1, pytorch ale potrebuje obzerváciu v tvare 1x80x80

ScaleFrame – Zmení škálovanie obzervácie.

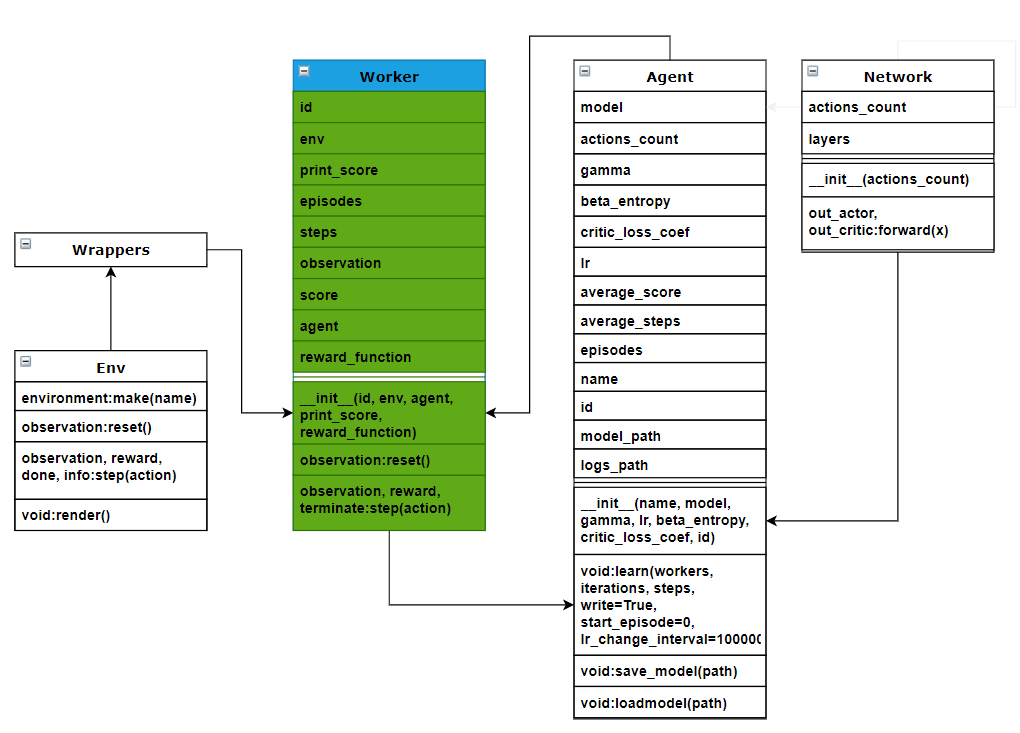
BufferWrapper(4) – Vytvorí skupinu 4 po sebe následujúcich obzervácií pre vytvorenie pohybu.

Po spracovaní má naša obzervácia tvar (4,80,80), kde 4 je počet po sebe nasledujúcich snímok pre znázornenie pohybu a 80,80 predstavujú výšku a šírku našej obzervácie.



Obrázok 2.3: Transformovaná obzervácia

## Pracovník

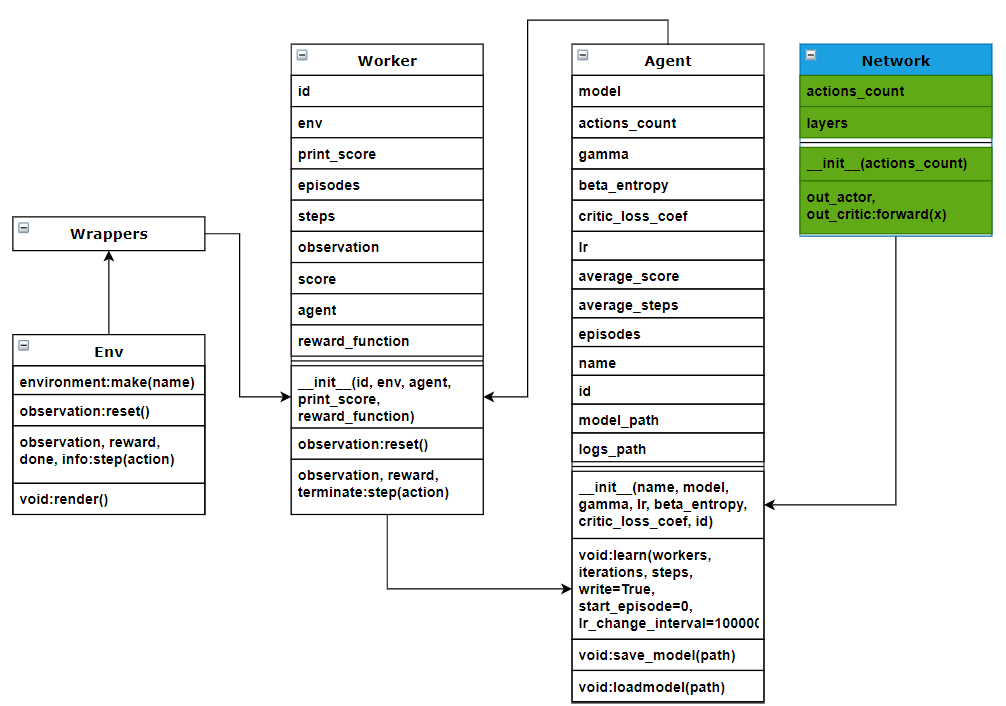


Obrázok 2.4: UML pracovníka

Pracovníci majú každý svoje id pre ich lepšie rozpoznanie, svoje vlastné prostredie v ktorom vykonávajú kroky a kópiu našeho agenta, prostredníctvom ktorého vykonávajú dané kroky. Ďalej obsahujú parametre na zbieranie údajov (vykonané epizódy, dosiahnuté skóre ...) a možnosť zadefinovať vlastnú odmenovú funkciu (reward function).

V našej práci nepoužívame priamo skóre z prostredia ale spracuvávame ho pomocou osobitnej odmenovej funkcie na hodnotu 1 alebo – 1 podľa toho či je naše skóre lepšie ako to predošlé alebo horšie. Tento prístup je výhodny pretože skóre môžu byť pomerne vysoké. To znamená vysoké zmeny pri nastavovaní parametrov, čo spôsobuje menej stabilný tréning a menej optimálne stratégie. Pri krokoch ktoré normálne negenerujú odmenu bude agent dostávať záporné odmeny. Aby nedostával záporné odmeny, musí si zvoliť cestu ktorá ho k odmene zavedie, nemôže teda robiť bezcieľne kroky. Príklady si ukážeme v časti trénovanie.

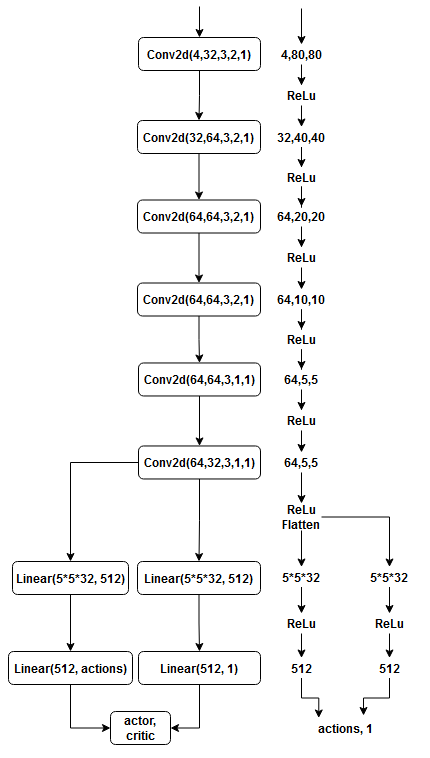
## Sieť



Obrázok 2.5: UML Neurónovej siete

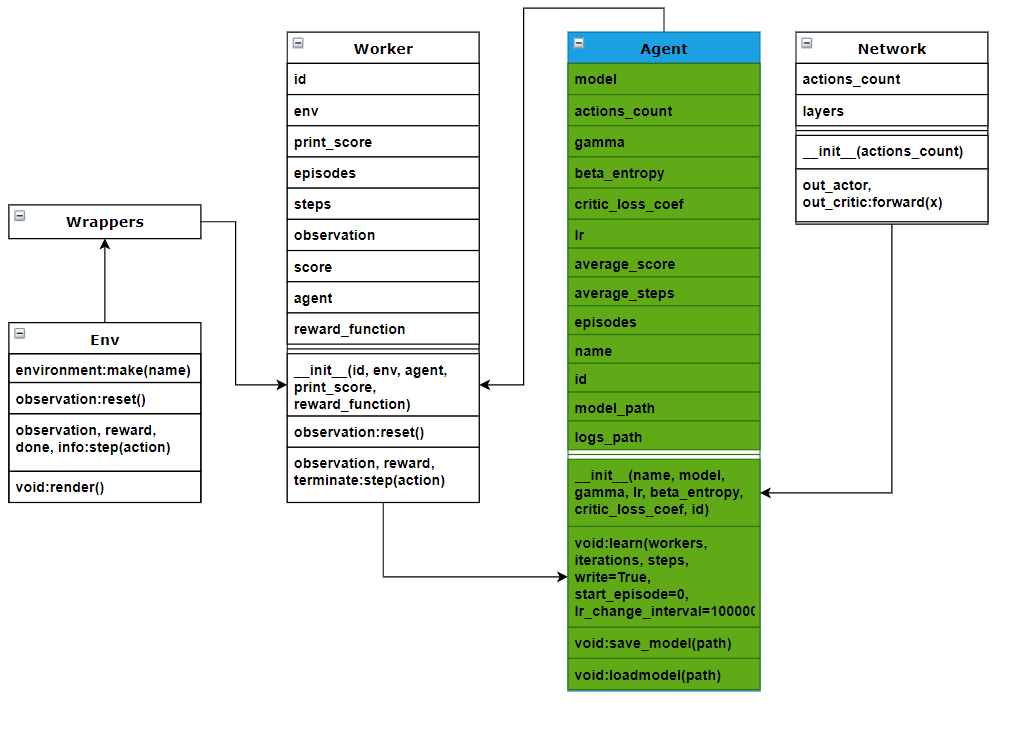
Pre vytvorenie našej neurónovej siete sme použili knižnicu Pytorch ktorá umožňuje jednoduchú tvorbu sietí pomocou modulu torch.nn. Pre spracovanie nášho prostredia používame konvolučné neurónové siete, ktoré extrahujú potrebné vlastnosti a zmenšia vstupné data. Výstup z konvolučných sieti smeruje do dvoch plne prepojených vrstiev. Prvá z nich predstavuje Actora a druhá Critica. Vyskúšali sme niekoľko dizajnov neurónových sieti ktoré budu bližšie popísane v časti trénovanie. Na obrázku nižšie môžete vidieť dizajn natrénovanej siete ktorú sme použili pre náš program.

Prvých 6 vrstiev sú konvolučné, z toho prvé 4 vrstvy zmenšia nášu vstupnú obzerváciu o polovicu a vykonajú aktivačnú funkciu ReLu. Ďalšie 2 vrstvy už rozmery nezmenšujú. Po konvolúcií nasleduje funkcia flatten ktorá nám vytvorí jednorozmerný vstup potrebný pre následujúce lineárne vrstvy. Lineárne vrstvy sú 4. Dve pre actora a dve pre critica. Výstupom sú naše požadované hodnoty. Pri vytvorení používame funkciu ktorá inicializuje parametre tejto siete na náhodné hodnoty v rozmedzí od 0 po 1.



Obrázok 2.6: Dizajn neurónovej siete

## Agent



Obrázok 2.7: UML Agenta

Agent obsahuje všetky hyperparametre a našu neurónovú sieť. Pri vytvorení sa pokúsi nájsť grafickú kartu. Pokiaľ ju nájde tak výpočty v priebehu učenia sa presunú na ňu. Pokiaľ grafickú kartu nenájde tak výpočty budú pokračovat na CPU. Agent obsahuje 3 metódy - save/load\_model pre ukladanie a načítavanie modelov počas trénovania a learn metódu ktorá je zodpovedná za učenie a ktorú si do hĺbky popíšeme.

## Učenie

### Implementácia

Učenie je implementované v metóde learn. Vstupné parametre pre túto metódu sú:

* Workers – Pole pracovníkov
* Iterations – Počet krokov agenta od začiatku až po krok optimalizátora a zapísanie štatistík.
* Steps – Počet krokov v prostredí. Tieto kroky sa všetky spracujú vrámcí jednej iterácie.
* Write – Parameter ktorý zapína ukladanie modelu a zbieranie dát.
* Start\_episode – Agenta je možné trénovať na viac krát. V prípade že zbierame dáta chceme vedieť nastaviť štartovaciu epizódu, aby sme vedeli nadviazať na predošle zozbierané dáta.
* Lr\_change\_interval – Počet epizód po ktorých zmeníme našu mieru učenia (lr).

Na začiatku trénovania sa zresetujú všetky prostredia aby sme získali vstupné obzervácie. Potom sa začína prvá iterácia. Na začiatku iterácie sa vytvoria premenné pre zbieranie potrebných údajov vrámci iterácie. Tieto premenné sú tensory tvaru [steps, workers, 1] a naplnia sa nulami. Po vytvorení premenných následujú kroky v prostredí. Vykonanie kroku pozostáva zo zavolania forward metódy našej neurónovej siete pre vygenerovanie actor a critic hodnôt. Actor hodnoty následne premeníme na pravdepodobnosti pre akcie v rozmedzí od 0 do 1 a logaritmy pre neskorší výpočet straty. Ďalej aktualizujeme naše premenné s údajmi za daný krok. V závere kroku vykonáme krok v prostredí pracovníkov. Obzervácie ktoré získame použijeme v ďalšom kroku. Po vykonaní všetkých krokov vrámci iterácie zavoláme forward nad poslednou sadou obzervácií aby sme získali critic hodnoty. Následne vypočítame advantage, straty a spravíme krok optimalizátora. V závere iterácie vypíšeme aktuálny stav, alebo zapíšeme dáta do súborov a uložíme model. //not sure ci dobre, mozno pridat code snippety ?

### Trénovanie

Prvotné parametre pre trénovanie boli poskytnuté našim školiteľom ktorý má skúsenosti s učením posilňovaním a A2C algoritmom.

Počet pracovníkov: 20

Počet iterácií trénovania: 5000

Počet krokov v prostredí: 10

Počet obzervácií pre jednu inštanciu prostredia: 4

Diskontný faktor: 0,99

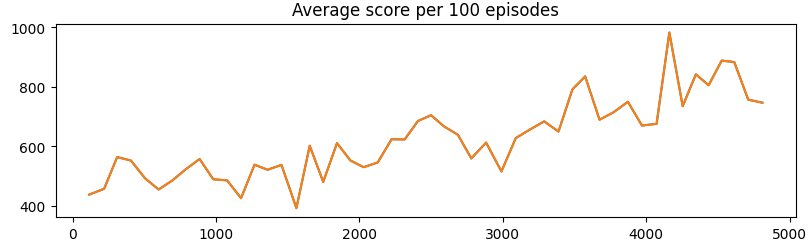
Miera učenia: 0,0005

Entropia: 0,01

Koeficient pre zníženie dôležitosti Critica: 0,5

Sieť: 4 konvolúčne vrstvy o veľkosti 64 -> plne prepojené vrstvy o veľkosti 512 neurónov -> výstupné plne prepojené vrstvy

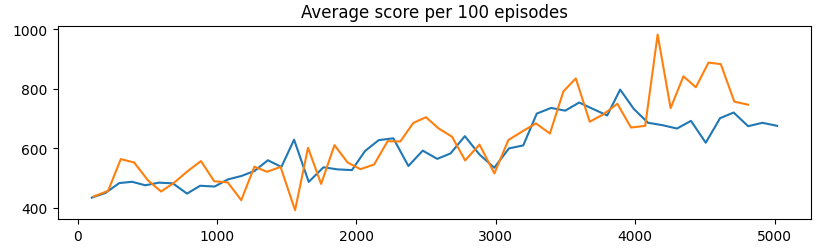
Výsledky po natrénovaní:



Obrázok 2.8: Prvotné skóre dosiahnuté prvým agentom

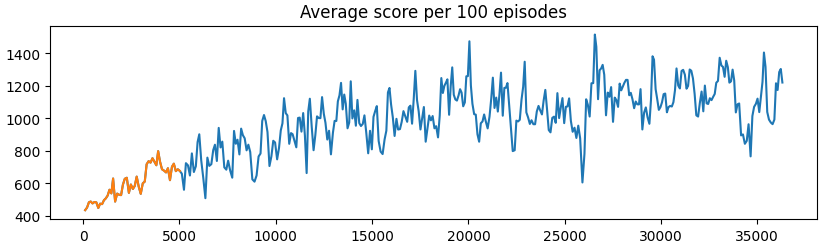
Zbierali sme priemerné skóre za posledných 100 epizód, aby bolo ľahšie rozoznávať trendy pri trénovaní. Zobrazený graf sa vzťahuje na vykonané epizódy, nie iterácie. Tento agent za 5000 iterácií vykonal 4810 epizód kde jedna epizóda predstavuje jednu hru. S aktuálnymi parametrami trvalo vykonanie 5000 iterácií približne 4 hodiny.

V ďalšom príklade sme upravili 2 parametre pre demonštráciu dôležitosti správneho nastavenia parametrov. Zvýšili sme počet krokov z 10 na 20 a znížili počet pracovníkov na 10. Čo znamená že dáta boli o kúsok viac korelované a agent sa aktualizoval po väčšom počte krokov. Tento test bol inšpirovaný článkom [12].



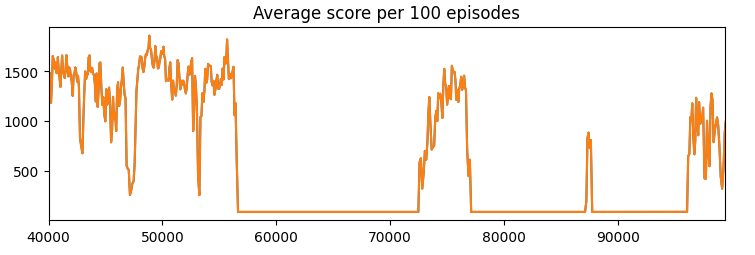
Obrázok 2.9: Priebeh tréningu s neoptimálnymi parametrami (modré) 1

Pri nižšom počte iterácií sa tento rozdiel prejaví ako pomalšie trénovanie, agent má ale stále potenciál pre zlepšenie.



Obrázok 2.10: Priebeh tréningu s neoptimálnymi parametrami 2

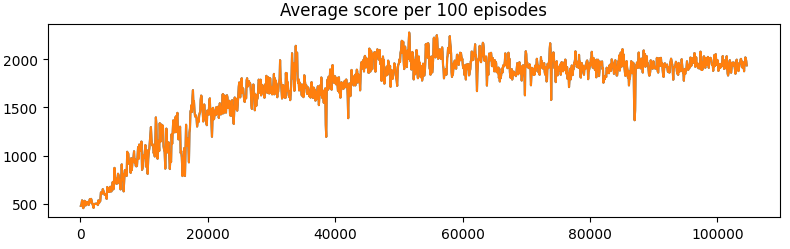
Pri dlhšom trénovaní môžeme vidieť vysoké kolísanie našeho skóre.



Obrázok 2.11: Priebeh tréningu s neoptimálnymi parametrami 3

Trénovanie je čím ďalej nestabilnejšie až do bodu keď sa agent odnaučí kompletne všetko. Od epizódy 57 000 vykonáva už len jeden úkon stále dookola a jeho dosiahnuté skóre je 90. Neskôr sa mu podarilo znova sa zlepšiť ale nedokázal prekonať svoje predošlé najlepšie skóre a „výpadky“ pokračovali naďalej.

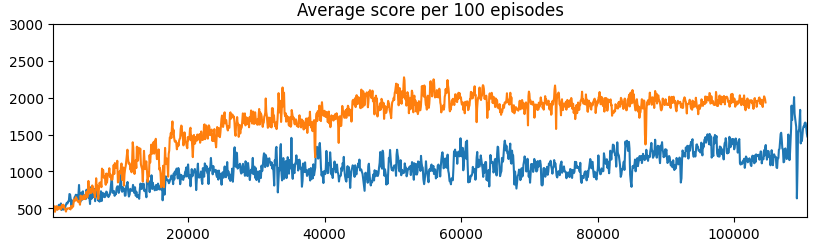
V trénovaní agenta sme pokračovali s našimi pôvodnými parametrami (20 pracovníkov a 10 krokov), ale mieru učenia sme znížili 5 násobne na hodnotu 0,0001 pre lepšiu stabilitu.



Obrázok 2.12: Najlepšie skóre dosiahnuté prvým agentom

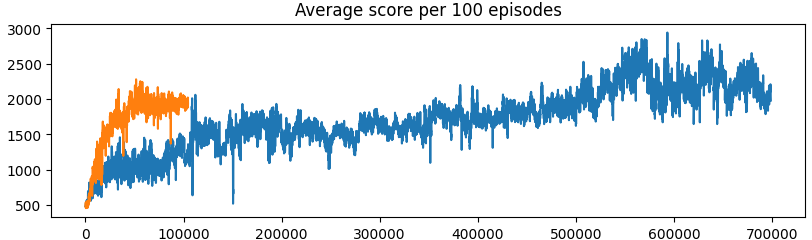
Nášeho pôvodného agenta sa nám podarilo vytrénovať na maximálne priemerné skóre okolo 2000. Pričom na strop možného skóre sme narazili okolo 50 000 epizódy čo je približne 62 000 iterácií. Od tohto bodu už agent nejavil žiadne zlepšenie.

Nakoľko na vyhratie hry potrebujeme minimálne skóre 2500 (získame pozbieraním všetkých „bodíkov“ roztrúsených po mape). Po prekonzultovaní s naším školiteľom sme pridali ďalšie 2 konvolučné vrstvy o veľkosti 64 pričom ostatné parametre sme nechali rovnaké ako predtým.



Obrázok 2.13: Porovnanie agent 1 (oranžová) vs agent 2 (modrá)

Pridaním dvoch vrstiev o veľkosti 64 sme zároveň pridali veľké množstvo nových parametrov ktoré treba nastaviť, preto agent na prvý pohľad vyzerá že sa mu darí horšie.



Obrázok 2.13: Kompletne natrénovaný agent 2

Len pridaním 2 konvolučných vrstiev sa počet epizód potrebných pre natrénovanie zväčšil skoro 10 násobne (uvažujeme strop prvého agenta pri 50 000 epizóde). Tento agent už dokáže vyhrať jednu hru (po výhre sa prostredie zresetuje a skóre ostáva). Natrénovanie tohto agenta trvalo približne xxx od zdenka na mašinke yyy.

Každá ďalšia zmena by zabrala príliž veľa trénovacieho času, preto v našej práci použijeme tohto agenta. Podľa Open AI najvyššie dosiahnuté skóre v prostredí Gym Pacman (dosiahnúte agentom) je 5738 a bolo dosiahnuté asynchrónnou implementáciou tohto algoritmu (A3C) [13].

Podľa Metodického usmernenia Čl. 3 Odporúčaná štruktúra záverečnej práce,  odsek (8) „*V časti* ***Súčasný stav riešenej problematiky*** *autor uvádza dostupné informácie a poznatky týkajúce sa danej témy. Zdrojom pre spracovanie sú aktuálne publikované práce domácich a zahraničných autorov*.“

***Podiel tejto časti práce má tvoriť približne 30 % práce.***

Príklad štýlu písania:

„*MDA je možné využiť aj v oblasti databázových systémov. V práci [24] autor využíva princípy MDA v oblasti DBRE , konkrétne v oblasti transformácie medzi relačnými databázami a objektovo orientovanými databázami. V práci sa ďalej uvádza, že automatizované prechody z jedného modelu do druhého budú realizované v zhode s MDA a definované prostredníctvom transformačných modelov na úrovni metamodelov. Na rozdiel od klasického MDA transformačného procesu, však nestačí transformovať len samotné modely, ale je potrebné zabezpečiť aj správnu migráciu dát, celý proces transformácie nie je možné podľa [27] realizovať v rámci jedného transformačné-ho kroku. Celý transformačný proces je zobrazený na obrázku 7.*“

**Záver kapitoly podrobne popisuje problém, ktorý nebol doteraz v danej téme výskumu riešený, alebo bol riešený iným prístupom a nami predpokladané riešenie by mohlo priniesť lepšie výsledky.**

# Ciele práce

Podľa Metodického usmernenia Čl. 3 Odporúčaná štruktúra záverečnej práce odsek (9) „Časť **Cieľ práce jasne, výstižne a presne charakterizuje predmet riešenia**. Súčasťou sú aj rozpracované čiastkové ciele, ktoré podmieňujú dosiahnutie cieľa hlavného.“

**Príklad spracovania cieľa**

„*Z analýzy súčasného stavu metodologických prístupov k vývoju IS a na základe analýzy prístupov k riešeniu transformácie CIM do PIM v MDA možno považovať za základný problém riešenia posunutie vývoja IS na čo najvyššiu úroveň abstrakcie. Dôvodom, prečo vývoj IS posúvať na čo najvyššiu možnú úroveň abstrakcie, je potreba čo najlepšej špecifikácie používateľských potrieb pre návrh informačného systému. Použi-tie princípov MDA prístupu pri návrhu IS slúži na vytvorenie základného piliera na preklenutie priepasti medzi doménou biznis analytikov a používateľov a doménou systémových analytikov a návrhárov. Zároveň v princípoch MDA architektúry je zárodok toho, ako povzniesť vývoj IS na čo najvyššiu úroveň abstrakcie, čo by mohlo viesť k odstráneniu závažných chýb, ktoré vznikajú pri tvorbe IS v súčasnosti*.

Preto sme si stanovili cieľ práce:

***Vytvoriť systémový prístup k transformácii CIM do PIM v MDA, podľa ktorého bude realizovaná transformácia modelu biznis procesov do vybraných návrhových modelov IS.***

*Dosiahnutie tohto cieľa vyžaduje rozčlenenie riešenia do štyroch parciálnych cieľov:*

1. *Existencia viacerých notácií na reprezentáciu formálneho zápisu modelu biznis procesov vnáša do riešenia problém výberu najvhodnejšej notácie. Preto prvý parciálny cieľ práce je:*

***Výber notácií umožňujúcich formálny zápis modelov biznis procesov, tak aby bola zrozumiteľná pre všetky zainteresované strany.***

1. *Úroveň PIM je reprezentovaná návrhovými modelmi v UML. Štandard UML poskytuje množstvo rôznych modelov na rôzne účely. Pre použitie v CIM – PIM transformácii nie sú všetky vhodné. Druhý parciálny cieľ je:*

***Špecifikácia UML modelov reprezentujúcich úroveň PIM.***

1. *Automatizovaná alebo polo – automatizovaná transformácia medzi modelmi nižších úrovní MDA je realizovaná na základe ich formálneho zápisu. Automatizovaná/polo – automatizovaná transformácia z CIM do PIM bude možná vtedy, keď budeme vedieť špecifikovať potrebný formálny zápis modelu biznis procesov. Pod formálnym zápisom je uvažovaný štandardizovaný grafický i sémantický (textový) formalizmus. To je úloha tretieho parciálneho cieľa:*

***Určenie požadovaného štandardizovaného formalizmu pre automatizovaný typ CIM – PIM transformácie.***

1. *Nájdenie vhodného problému a modelovanie jednotlivých modelov je podklad pre praktické overenie navrhovaných transformačných postupov. Overenie však bude iba čiastočné, lebo práca sa zameriava iba na najvyššie úrovne abstrakcie a transformácie medzi nimi. Pre úplné overenie by bolo potrebné previesť transformácie až po samotný zdrojový kód.*

***Posledným parciálnym cieľom je vytvorenie prípadovej štúdie, ktorá bude overením nami navrhovaného prístupu k CIM – PIM transformácii v MDA.*“**

# Metodika práce a metódy skúmania

Podľa Metodického usmernenia Čl. 3 Odporúčaná štruktúra záverečnej práce, odsek (10) „*Časť Metodika práce a metódy skúmania spravidla obsahuje:*

1. *charakteristiku objektu skúmania,*
2. *pracovné postupy,*
3. *spôsob získavania údajov a ich zdroje,*
4. *použité metódy vyhodnotenia a interpretácie výsledkov,*
5. *štatistické metódy.“*

**Poznámky pre spracovanie kapitoly**

**Metodika je podľa Wikipedie** – „***Náuka o spôsobe vedeckej práce. Je to súhrn pracovných spôsobov a metód v určitej oblasti.*** *Je to uvedomelý a cieľavedomý* [*postup*](https://sk.wikipedia.org/wiki/Postup)*, určitým spôsobom* [*usporiadaná*](https://sk.wikipedia.org/wiki/Usporiadanos%C5%A5)[*činnosť*](https://sk.wikipedia.org/wiki/%C4%8Cinnos%C5%A5) *alebo usporiadanie* [*operácií*](https://sk.wikipedia.org/wiki/Oper%C3%A1cia)*, pretvárajúcich* [*východiskové danosti*](https://sk.wikipedia.org/wiki/V%C3%BDchodiskov%C3%A1_danos%C5%A5) *istej* [*cieľavedomej činnosti*](https://sk.wikipedia.org/wiki/Cie%C4%BEavedom%C3%A1_%C4%8Dinnos%C5%A5) *na jej zamýšľaný (čiastočne alebo úplne realizovaný)* [*cieľ*](https://sk.wikipedia.org/wiki/Cie%C4%BE)*.“*

Záverečné práce sú súčasťou výskumnej a vývojovej činnosti univerzity, ktorou sa realizuje vedecká práca.

Spracovanie metodiky je závislé od:

* riešenej témy,
* mechanizmu riešenia,
* použitých metód,
* spôsobu overenia,
* .....

Vzorom môže byť základný životný cyklus vývoja systémov z fázami: analýza/rozbor, návrh a implementácia/realizácia.

**Dôležitosť metodiky** je v jej užitočnosti. Pre riešiteľa má byť hlavne:

* Spôsobom, ako systémovo a zmysluplne postupovať v riešení.
* Vzbudiť dôveru k dosiahnutiu cieľov.
* Odstrániť pocit ťažkosti/prísnosti vedeckých prístupov.

Ak riešiteľ aj napriek vyššie uvedeným dôvodom nepotrebuje alebo nevie vytvoriť metodiku vedeckovýskumnej práce, nemusí ju spracovať ešte pred riešením. Metodika riešenia je však dôležitá pri záverečnom zhodnotení výsledkov. Treba ju dodatočne spracovať, aj keď postup riešenia bol ad hoc. Je potrebná pre posudzovanie správnosti riešenia je aj východiskom pre riešenie podobných výskumných problémov.

Podľa Wikipédie„***Metóda je poznaný*** [***zákon***](https://sk.wikipedia.org/wiki/Z%C3%A1kon) ***premenený na*** [***pravidlo***](https://sk.wikipedia.org/wiki/Pravidlo)***,*** [***súbor***](https://sk.wikipedia.org/wiki/S%C3%BAbor)[***pravidiel***](https://sk.wikipedia.org/wiki/Pravidlo)***, systém regulatívnych princípov.“***

Spracovanie použitých metód v dizertačnej práci môže byť založené na nasledujúcej štruktúre:

* Metódy vedeckého poznania, kde patria Logické metódy
  + analýza,
  + syntéza,
  + abstrakcia,
  + konkretizácia,
  + indukcia,
  + dedukcia
  + .....
* Metódy vedeckého skúmania
  + empirické skúmanie (pozorovanie, meranie, experiment,...)
  + logické spracovanie empirických údajov (klasifikácia, typologizácia, korelácia,...),
  + ....

# Výsledky Práce

## Riešenie problému

Podrobný popis postupov podľa metodiky riešenia.

## Výsledky riešenia

Výsledky, ktoré boli dosiahnuté riešením.

Podľa Metodického usmernenia Čl. 3 Odporúčaná štruktúra záverečnej práce, odsek (11) *„****Výsledky práce a diskusia sú najvýznamnejšími časťami záverečnej práce****. Výsledky (vlastné postoje alebo vlastné riešenie vecných problémov), ku ktorým autor dospel, sa musia logicky usporiadať a pri popisovaní sa musia dostatočne zhodnotiť. Zároveň sa komentujú všetky skutočnosti a poznatky v konfrontácii s výsledkami iných autorov.* ***Ak je to vhodné, výsledky práce a diskusia môžu tvoriť aj jednu samostatnú časť a spoločne tvoria spravidla 30 až 40 % záverečnej práce.“***

**Podľa uváženia spracujte zvlášť kapitolu Diskusia výsledkov hlavne vtedy, ak budete porovnávať výsledky práce s inými riešeniami.**

# Záver

Podľa Metodického usmernenia Čl. 3 Odporúčaná štruktúra záverečnej práce, odsek (12) *„V závere je potrebné v stručnosti zhrnúť dosiahnuté výsledky vo vzťahu k stanoveným cieľom.“*

# Zoznam použitej literatúry

**V aktuálnom dokumente neexistujú zdroje.**

Podľa Metodického usmernenia Čl. 3 Odporúčaná štruktúra záverečnej práce, odsek (14)

*„Zoznam použitej literatúry obsahuje úplný zoznam bibliografických odkazov. Rozsah tejto časti je daný* ***množstvom použitých literárnych zdrojov, ktoré musia korešpondovať s citáciami použitými v texte.****“*

Norma STN ISO 690 z mája 2012 poskytuje návod na tvorbu bibliografických odkazov a citácií, nepredpisuje však ich konkrétny štýl. Norma stanovuje, že každý údaj v odkaze musí by zreteľne oddelený. Zároveň odporúča poradie údajov v odkaze, uvádza povinné a voliteľné údaje potrebné pre identifikáciu odkazovaného. Príklady použitia štýlov bibliografických odkazov:

**Knižné publikácie (monografie, učebnice, zborníky...)**

OBERT, V., 2006. *Návraty a odkazy.* Nitra: Univerzita Konštantína Filozofa. ISBN 80-8094-046-0.

ROHOŇOVÁ, Ľudmila a Mária GAŠPAROVÁ, 2006. *Metodika popisu článkov vo formáte MARC 21 s interpretáciou pravidiel AACR: príručka používateľa: verzia V 2.1.* [online]. Martin: Slovenská národná knižnica [cit. 2013-01-24]. ISBN 80-89023-69-X. Dostupné z: http://www.snk.sk/nbuu/metodika\_clanky\_MARC21.pdf

**Článok v časopise**

ANDREJČÍKOVÁ, Nadežda, 1999. Komunikácia a kooperácia IS pre knižnice: úvod do komunikačného protokolu Z39.50. In: *Bulletin Centra vedecko-technických informácií SR*. Roč. 3, č. 2, s. 54-59. ISSN 1335-793X.

KUCIANOVÁ, Anna, 2001. Významná personálna bibliografia. In: *Knižnica* [online]. Roč. 2, č. 11-12, s. 662 [cit. 15. júna 2013]. ISSN 1336-0965. Dostupné z:

http://www.snk.sk/swift\_data/source/NBU/Kniznica/11\_12\_2001/662.pdf

**Článok (kapitola) z knižnej publikácie (zborník, monografia, učebnica...)**

KOMOROVÁ, Klára, 2005. Najstaršie trnavské tlače v Slovenskej národnej knižnici v Martine. In: *Kniha 2005: zborník o problémoch a dejinách knižnej kultúry*. Martin: Slovenská národná knižnica, s. 341-345. ISBN 80-89023-55-X.

KUCIANOVÁ, Anna, 2005b. Personálie v elektronickej súbežnej Slovenskej národnej bibliografii. In: *Bibliografický zborník 2000-2001* [online]. Martin: Slovenská národná knižnica, s. 136-139 [cit. 24. septembra 2012]. Dostupné z: http://www.snk.sk/nbuu/Zborník\_2000\_2001.pdf

**Vedecko-kvalifikačné práce**

MIKULÁŠIKOVÁ, M., 1999. *Didaktické pomôcka pre praktickú výu*č*bu na hodinách výtvarnej výchovy pre 2. stupe*ň *základných škôl*: *diplomová práca*. Nitra: UKF.

**Výskumné správy**

BAUMGARTNER, J. a kol., 1998. *Ochrana a udržiavanie genofondu zvierat, šľachtenie zvierat*: *výskumná správa*. Nitra: VÚŽV.

**Normy**

ISO 690: 2010*, Information and documentation. Guidelines for bibliographic references and citations to information resources.*

**Archívne materiály**

BANÍK, Anton Augustín, 2010. *Korešpondencia s Pavlom Horváthom*. 2 s. Rukopis. Metodická diskusia z r. 1911. Umiestnenie: Martin: SNK, Archív literatúry a umenia. Signatúra A XC/ 4-1.

Informácia z WWW

Convention on biological diversity : Article 18. Technical and Scientific Cooperation.  
<http://www.biodiv.org/convention/articles.asp> (2001-09-28)

Prílohy

## Príloha A:

## Príloha B: Obsah DVD

Priložené DVD obsahuje:

* Práca v elektronickej podobe (formát PDF)

# Ďalšie usmernenia pre písanie záverečnej práce

Podľa Metodického usmernenia Čl. 5 **Odporúčaná formálna úprava záverečných prác odsek (1) „*Formálna úprava záverečných prác vychádza z noriem:***

*STN ISO 2145: 1997. Dokumentácia. Číslovanie oddielov a pododdielov písaných dokumentov.,*

*STN ISO 214: 1998, Dokumentácia. Abstrakty (referáty) pre publikácie a dokumentáciu.,*

*STN ISO 690: 1998. Dokumentácia – Bibliografické odkazy – Obsah, forma a štruktúra.,*

*STN 01 6910: 1999. Pravidlá písania a úpravy písomností.,*

*STN ISO 690-2: 2001, Informácie a dokumentácia. Bibliografické citácie. Časť 2: Elektronické dokumenty alebo ich časti."*

**odsek (3) „Odporúčaný typ písma** je Times New Roman, veľkosť 12 a je jednotný v celej práci. Odporúčané nastavenie strany - riadkovanie 1,5; okraje zľava 3,5 cm, sprava 2 cm, zhora a zdola 2,5 cm, orientácia na výšku, formát A4.

## ČLENENIA KAPITOL A PRÁCE S ILUSTRÁCIAMI, TA-BUĽKAMI, ROVNICAMI A KRÍŽOVÝMI ODKAZMI

V práci sa môžu vyskytovať okrem slovného textu aj informácie vyjadrené v obrazovej forme a symbolmi.

### Názov tretej úrovne

Editujte svoju prácu v kapitolách a podkapitolách. Čísla kapitol a podkapitol (druhej a tretej úrovne) sa citujú v texte práce takto:

V kapitole 1 sme už uviedli, že ...; ... pozri 2.1 ... atď. ...

Odporúčaný rozsah bakalárskej práce je 30 až 40 strán (54 000 až 72 000 znakov vrátane medzier), diplomovej práce 50 až 70 strán (90 000 až 126 000 znakov), dizertačnej práce 80 až 120 strán (144 000 až 216 000 znakov) a habilitačnej práce do 150 strán.

Do tohto rozsahu sa počíta len hlavný text, t. j. úvod, kapitoly, záver a zoznam použitej literatúry. Dôležitejší ako rozsah práce je kvalita práce a úroveň jej spracovania. Pri písaní je dôležité dbať na vyváženosť (proporcionálnosť) jednotlivých častí práce:

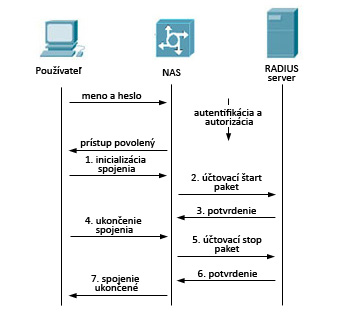
* úvod má spravidla 2 – 3 strany,
* teoreticko-metodologická časť tvorí spravidla jednu tretinu práce,
* ostatné kapitoly tvoria približne dve tretiny práce,
* záver má zvyčajne 2 – 3 strany.

## Ilustrácie

Ilustrácie sú obrázky obsahujúce grafy, diagramy, mapy, schémy a pod. Nie je potrebné rozlišovať rozličné typy ilustrácií, stačí, ak sa všetky označia ako „Obrázok”. Všetky ilustrácie budú očíslované číslom kapitoly a súvislým radom číslic v danej kapitole práce a musia mať titulky (názov obrázku) pri každom obrázku. Text titulku musí byť pochopiteľný aj bez kontextu (t.j. ak je obrázok prvý v kapitole 1, číslo obrázka bude 1.1, ak je obrázok tretí v kapitole 4,, číslo obrázka bude 4.3 a pod.). Obrázky prekreslite do jednotného formátu, vkladanie rôznych výsekov z literatúry nie je vhodné. Odporúčam použiť niektorý z voľne šíriteľných softvérov, alebo MS Visio (na FRI je možné k nemu získať prístup).

Obrázky sa majú zaradiť bezprostredne za textom, kde sa spomínajú po prvý-krát (najlepšie na tej istej strane). Obrázok by mal byť podľa možnosti centrovaný. Pri odkazovaní na daný obrázok v texte treba použiť krížové odkazy na obrázok (napr. Obr. Obrázky a tabuľky vkladáme v tejto šablóne takto:

1. Vložiť → Obrázok
2. Pravý klik na vložený obrázok → Vložiť popis
3. V okne Popis pri možnosti Označenie vyberieme „Obr.“
4. Pri možnosti Umiestnenie vyberieme možnosť „Pod vybratou položkou“
5. Do kolónky Popis dopíšeme názov obrázku
6. Štýl „ZP\_Popis obrázka / tabuľky“



Obrázok .: Účtovanie pri protokole RADIUS.

V texte sa na vytvorené obrázky a tabuľky odkazujeme krížovými odkazmi (pozri 2.4).

1. Krížový odkaz na obrázky a tabuľky vytvoríme takto:
2. nastavíme sa kurzorom tam, kde sa má objaviť odkaz,
3. z panela nástrojov vyberieme: Vložiť (Insert) → Krížový odkaz (Cross-reference),
4. v dialógovom boxe pre Krížový odkaz vyberieme v poli Typ odkazu (Reference Type) vyberieme Tab. alebo Obr.,
5. v poli Vložiť odkaz (Insert reference) vyberieme vložiť odkaz na: Only label and number (popis a číslo),
6. zo zoznamu popisov vyberieme tabuľku alebo obrázok a zaškrtneme Hypertex-tový odkaz (Insert as Hyperlink),
7. Insert a Close.
8. Aktualizácia krížových odkazov v celom dokumente: **CTRL + A**, potom **F9**.

## Tabuľky

Tabuľky prezentujú myšlienky a tvrdenia popisované v práci. Akýkoľvek tabuľkový materiál, ktorý sa skladá z viac než štyroch alebo piatich riadkov, by mal byť spracovaný do formy tabuľky, napríklad takej ako v je v Tab. 2.1. Popis a záhlavie tabuľky má byť zrozumiteľné samostatne bez odkazu na text. Záhlavia majú vyjadrovať druh veličiny a typy jednotiek vo forme „veličina/jednotka”, je potrebné používať rovnaké symboly a skratky ako v texte. Každá tabuľka musí mať poradové číslo a titulok, umiestnený zvyčajne nad tabuľkou. Tabuľka by mala mať rovnakú orientáciu, ako text práce. Tabuľka by mala byť číslovaná ako obrázky a grafy.

|  |  |  |
| --- | --- | --- |
| ORGANIZAČNÉ SCHÉMY  (usporiadanie informácií) | | ORGANIZAČNÉ ŠTRUKTÚRY  (hierarchizácia informácií) |
| Exaktné | Neexaktné | Hierarchia zhora dole |
| Alfabetické | Tematické | Databázový model zdola hore |
| Chronologické | Podľa publika | Hypertext |
| Geografické | Metaforické | Sociálna klasifikácie |
|  | Hybridné |  |

Tabuľka : Organizačné schémy a štruktúry IA (Morville – Rosenfeld, 2006)

## Rovnice, vzorce

Rovnice sa uvádzajú v strede riadka, vysvetlivky symbolov na začiatku riadku. Vysvetlivky symbolov sa uvádzajú od začiatku riadka. Ak je v práci viac vzorcov, uvá-dzame číslo vzorca do okrúhlych zátvoriek bez medzier umiestnených na pravom konci riadka. Pre písanie fyzikálnych veličín a matematických premenných sa používa kur-zíva. Používame sústavu jednotiek SI (ISO 31 a ISO 1001). Pri písaní rovníc používame editor rovníc (musíme ho mať nainštalovaný).

## Krížové odkazy

Pomocou funkcie krížový odkaz je možné vytvárať odvolávky v texte na kapito-ly, podkapitoly , rovnice, vzorce, obrázky, tabuľky ;a pod. Majú formu: rovnica (2), pozri Katuščák (1996), ako sme uviedli v kapitole 1 a pod.

Krížové odkazy je možné vytvoriť pre číslované zoznamy, nadpisy, záložky, po-známky pod čiarou, popisy, číslované odseky a pod. Základom je mať vytvorený číselný zoznam referencií. Potom je možné vytvoriť krížový odkaz na tieto referencie, pričom keď sa zmení poradie referencií v zozname, Word vie automaticky aktualizovať čísla referencií. **Aktualizácia krížových odkazov v celom dokumente sa vykoná cez klávesy CTRL + A potom F9.**

## Skratky a odborné pojmy

Odborné anglické pojmy uveďte v práci pri prvom použití, a ak sa dá, uveďte za pojmom do zátvorky skratku a ďalej v texte používajte len ju (Príklad: Access point (AP)). Nezabudnite na začiatku práce uviesť zoznam všetkých v práci použitých skratiek. Zoznam skratiek abecedne utrieďte.

V prípade, že existuje zaužívaný slovenský preklad anglického pojmu použite v práci slovenský, napríklad smerovač (router), prepínač (switch).

## Odrážky

Pri použití číslovaných aj nečíslovaných odrážok použite zadefinované štýly.

* Prvá úroveň sú plné okrúhle.
  + Druhá úroveň sú plné hranaté.

## Gramatika

Vysoko odporúčam zapnúť a používať pri kontrole gramatiky vstavané mechanizmy kontroly gramatiky vo Worde (F7).

## Obhajoby

Pre obhajoby platia všeobecné pravidla, dostupné vo vloženom PDF súbore.

Na stránke katedry je popísané odporúčanie k tvorbe prezentácii na obhajobu. Adresa je: http://www.kis.fri.uniza.sk/sk/studenti-a-absolventi/odporucania/232-ako-pisa-prezentaciu-na-obhajobu-bc-ing

Pri písaní teda platia aj nesledujúce pravidlá:

* voľte vhodnú šablónu, aby bola na projektore dobre viditeľná (vyhýbajte sa krik-ľavým farbám a rôznym čiaram na pozadí textu), na druhu stranu iba biely slide nie je nič moc,
* slajdy musia byť heslovité, t.j. riešené cez odrážky, žiadne dlhé kópie textu ale-bo odsekov z prace,
* obsah slajdu prezentujete, komentujete, avšak nečítate ho mechanicky z projek-tora, preto si pripravte k slajdom text alebo komentár prejavu. Pomôže vám to potom aj pri samotnej prezentácii,
* dodržujte čas prezentácie, napr. na 12 minút prezentácie alebo je odhadom max. 12 slajdov, kde na prvom by ste sa mali predstaviť a na poslednom jasne ukončiť prezentáciu. Vysoko odporúčam prezentáciu prejsť niekoľkokrát pred tým ako vystúpite pre komisiou a vyskúšať si časovanie a prednes,
* pripravte si na ppt aj otázky vedúceho a oponenta.
* prezentujte čo ste v DP/BP robil a nevyučujte komisiu zbytočnou teóriou
* z prezentácie musí byt jasne čo ste riešil a vyriešil vo vašej DP/BP
* PowerPoint aj LibreOffice po stlačení klávesy F7 vám prekontrolujú gramatiku!!!!!!! btw..ma to aj word

[1] <https://sk.wikipedia.org/wiki/Moorov_z%C3%A1kon>

[2] <https://www.expert.ai/blog/machine-learning-definition/>

[3] <https://arxiv.org/pdf/1509.06461.pdf>

[4] <https://arxiv.org/pdf/1511.06581.pdf>

[5] <http://blog.shakirm.com/2015/11/machine-learning-trick-of-the-day-5-log-derivative-trick/>

[6] <https://openai.com/blog/baselines-acktr-a2c/>

[7] <https://openai.com/blog/openai-baselines-ppo/>

[8] <https://arxiv.org/pdf/1707.06347.pdf>

[9] <https://marutitech.com/top-8-deep-learning-frameworks/> , <https://analyticsindiamag.com/deep-learning-frameworks/>,

<https://365datascience.com/trending/deep-learning-frameworks-2019/>

[10] <https://www.toptal.com/machine-learning/tensorflow-machine-learning-tutorial>

[11] <https://pytorch.org/features/>

[12] <http://blog.jzhanson.com/blog/rl/project/2018/05/28/breakout.html>

[13] <https://gym.openai.com/evaluations/eval_kpL9bSsS4GXsYb9HuEfew/>

<https://github.com/europa502/RNN-based-Bitcoin-Value-Predictor> obr 1.5