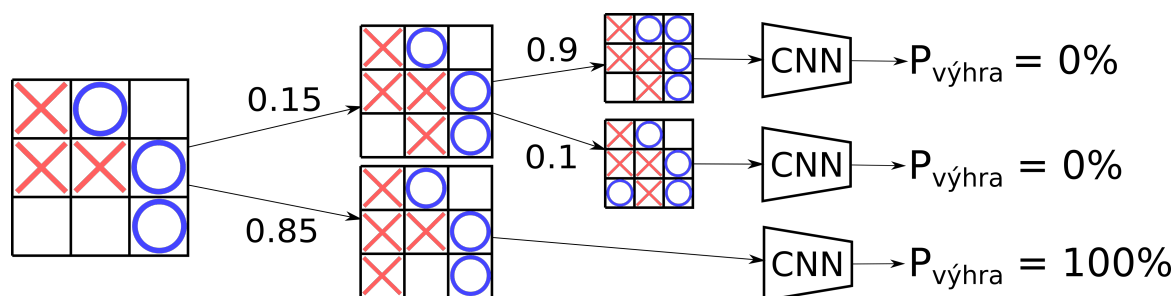


Učiaca sa umelá inteligencia pre piškvorky

Matúš Bako



Abstrakt

V mojej práci sa zaoberám vytvorením umelej inteligencie, ktorá dokáže hrať hru gomoku, ktorá má veľmi podobné pravidlá ako hra Piškvorky. Narozdiel od konvenčných metód prehľadávania stavového priestoru a ručne definovaných pravidiel kombinujem stochastické prehľadávanie s konvolučnými neurónovými sieťami. Strategická neurónová sieť určuje najpravdepodobnejší ťah pri danej konfigurácii a ohodnocovacia sieť určuje pravdepodobnosť výhry. Vyhodnotil som rôzne architektúry sietí a rôzne spôsoby tréningu. Z experimentov vyplnilo, že bez použitia ohodnocovacej siete je problematické zakončiť hru. Napriek relatívne malej množine tréningových dát je mnou vytvorená umelá inteligencia schopná konkurovať nižšie umiestneným umelým inteligenciám zo súťaže Gomocup.

Kľúčové slová: Gomoku — Neurónové siete — Umelá inteligencia

Priložené materiály: [Gomocup](#) — [Zdrojový kód](#)

xbakom01@stud.fit.vutbr.cz, Fakulta informačných technológií, Vysoké učení technické v Brne

1. Úvod

Stolné hry sú niekoľko desaťročí terčom strojového učenia, pretože majú jasne definované pravidlá a kritériá výhry. Napriek tomu mnoho z nich poskytuje veľký počet dosiahnuteľných konfigurácií, čo vytvára priestor pre súťaživosť a zdokonaľovanie. Pre hľadanie správneho ťahu sú používané rôzne metódy, napríklad prehľadávanie stavového priestoru, prípadne rôzne druhy učení. Líšia sa hlavne časovou a priestorovou zložitou.

V mojej práci sa snažím demonštrovať použiteľnosť konvolučných neurónových sietí pri hraní hry gomoku. Konvolučné neurónové siete už boli použité na vytvorenie umelej inteligencie pre hranie hier, medzi ktoré patria šach, Go, Backgammon a iné. Narozdiel od tradičných programov, ktoré majú natvrdo naprogramované

isté pravidlá a heuristiky, používam na výber nasledujúceho ťahu neurónové siete.

Piškvorky je veľmi jednoduchá stolná hra s jasne definovanými pravidlami a podmienkou výhry. Je to hra s nulovým súčtom, úplnými informáciami, je určená pre dvoch hráčov a nie je založená na náhode. Jej nevýhodou ale je malý stavový priestor. Preto som pre experimentovanie zvolil inú, ale zároveň veľmi podobnú, hru gomoku. Jediný rozdiel v základných pravidlách je vo veľkosti hracej plochy. Niektoré varianty počítajú s pätnástimi riadkami a stĺpcami, avšak ja som zvolil variantu, kde ich je devätnásť. Hráč vyhráva, ak má na hracej ploche v rade päť kameňov jeho farby v ortogonálnom alebo diagonálnom smere.

Problémom, ktorý je podobný rozoberanému problému v tejto práci, sa zaoberala spoločnosť DeepMind. Ich cieľom bolo vytvoriť umelú inteligenciu

Typ siete	Okolie	Konvolučné vrstvy			
L2	7	5	3		
		256	1		
L3	7	3	3	3	
		64	128	1	
L4	13	3	5	5	3
		64	64	32	1
L5	15	3	5	5	3
		32	64	64	32
L6	13	3	3	3	3
		32	64	64	32

Tabuľka 1. Popis architektúr jednotlivých strategických sietí. Druhý stĺpec reprezentuje rozmer plochy, ktorá ovplyvňuje každé políčko hracej plochy. Horný riadok pravej časti tabuľky obsahuje veľkosť konvolučného jadra a spodný počet výstupov vrstvy.

hrajúcu hru Go. Táto hra sa hrá na rovnako veľkej hracej ploche ako Gomoku, avšak pravidlá sú komplexnejšie a zakladajú sa na ohraničovaní plochy pomocou kameňov. Nakoniec sa im podarilo vytvoriť umelú inteligenciu zvanú AlphaGo [1], ktorá porazila európskeho šampióna so skóre 5:0. Bolo to prvý krát, kedy umelá inteligencia porazila profesionálneho hráča Go. Ich riešenie si zakladá na použití dvoch neurónových sietí a ich zakomponovaní do vyhľadávacieho algoritmu. Jedna z neurónových sietí, zvaná ako strategická sieť, dostane na vstup aktuálnu konfiguráciu hracej plochy a jej výstupom sú pravdepodobnosti hrania na danú pozíciu priradené príslušajúcim pozíciám na hracej ploche. Druhá neurónová sieť dostáva na vstup konfiguráciu hracej plochy a jej výstupom je pravdepodobnosť výhry pri danej konfigurácii. Obe neurónové siete sú využívané pri stochastickom prehľadávaní stavového priestoru. Medzi metódy, ktoré využili pri tréningu, taktiež patrí posilňované učenie [2]. Tým sa umelá inteligencia naučila hrať sama proti svojim starším verziám, čo zlepšilo celkovú kvalitu hrania.

Moje riešenie sa inšpiruje z AlphaGo a tiež používa dve neurónové siete. Rozdiel je ale v prehľadávaní stavového priestoru. AlphaGo využíva vyhľadávací algoritmus založený na metóde stromového prehľadávania Monte Carlo. Ja som sa ním inšpiroval a prispôbil som si ho využitím strategickú a ohodnocovú sieť. Pomocou nej z prvej konfigurácie hracej plochy generujem nasledujúce prvotné ťahy, čím vytváram nasledujúce konfigurácie. Ohodnocovacia sieť potom priradzuje pravdepodobnosť výhry konkrétnym konfiguráciám. Priemerom všetkých pravdepodobností je pravdepodobnosť výhry pri zvolení po-

program : sieť	L2	L3	L4	L5	L6
Imro	96:4	94:6	50:50	63:37	50:50
	60:20	85:15	58:42	58:42	50:50
Mentat	93:7	87:12	55:45	47:53	30:70
	94:6	88:11	50:50	49:51	45:55
Gmotor	98:2	95:5	93:7	88:12	90:10
	98:2	96:4	91:9	91:9	93:7

Tabuľka 2. Výsledky hier stiahnutých programov proti strategickú sieť. Na hornom riadku z dvojice je výsledok so sieťou natrénovanou na všetkých ťahoch a na spodnom výsledok siete trénovananej iba na ťahoch, ktoré viedli k výhre. Imro je v tabuľke súťaže Gomocup na 32. , Mentat na 27. a Gmotor na 12. mieste.

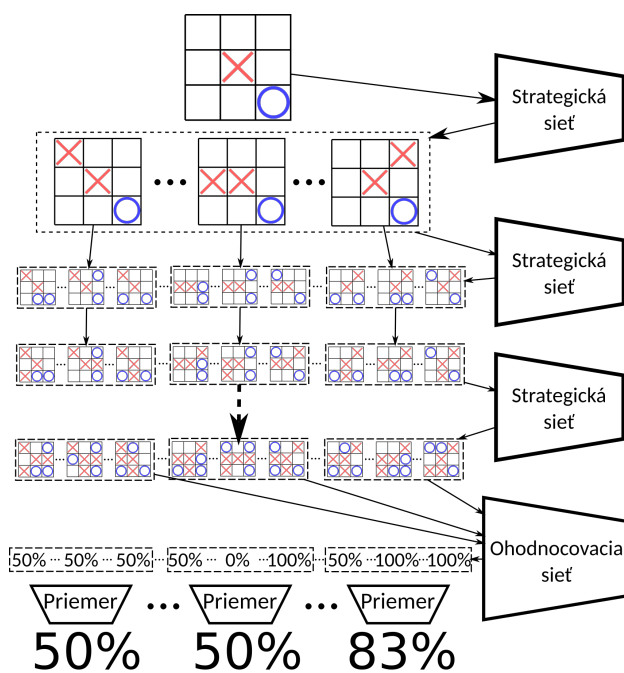
čiatocného ťahu.

Narozdiel od konvenčných programov, ktoré používajú ručne definované pravidlá, sme dokázali natrénovať strategickú sieť, ktorá sama o sebe dokáže poraziť nižšie umiestnené programy zo súťaže Gomocup. Sieť je možné pretrénovať aj na iné varianty, ako napríklad Renju. Jedinou podmienkou je získať dostatočný počet unikátnych hier.

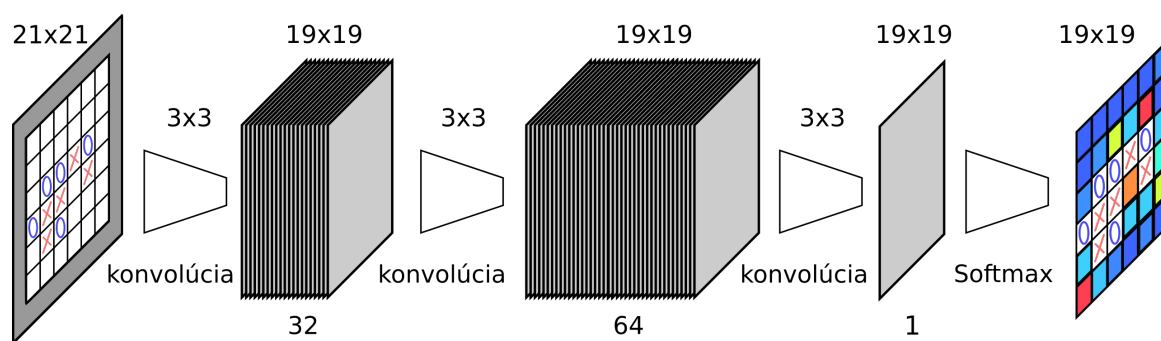
2. Metodológia

2.1 Vyhodnocovanie ťahu

Pre určenie nasledujúceho ťahu strategická sieť dostane na vstup konfiguráciu hracej plochy a jej výstupom je



Obrázok 1. Prehľadavací algoritmus ktorý využíva strategickú a ohodnocovú sieť pre určenie nasledujúceho ťahu.



Obrázok 2. Architektúra strategickej siete s tromi konvolučnými vrstvami. Vstupná hracia plocha je ohraničená výplňou pre vyznačenie hranice plochy a zachovanie rozmerov. Rovnako je výplň použitá na ohraničenie plochy aj v každej konvolučnej vrstve. Výstupom je pravdepodobnosť ťahu pre všetky políčka ilustrovaná teplotnou mapou. Pôvodné žetóny sú na mape ponechané iba kvôli ilustrácii hodnôt okolitých políček.

pravdepodobnostné rozloženie pre všetky pozície na ploche. Ohodnocovacia sieť dostane na vstup konfiguráciu hracej plochy a výstupom je pravdepodobnosť výhry pri danej konfigurácii.

Vstupom oboch sietí je pole s dvomi kanálmi, pričom prvý kanál reprezentuje vlastnú hraciu plochu a druhý kanál hraciu plochu protivníka. Zabrané miesto v poli má hodnotu 1 a voľné políčka sú označené hodnotou 0. Jednotlivé kanály reprezentujú pole o veľkosti 19x19 ohraničené výplňou s hodnotou -1. Šírka výplne je vypočítaná pomocou vzorca $\lfloor k/2 \rfloor$, kde k je veľkosť konvolučného jadra prvej konvolučnej vrstvy.

2.2 Vyhľadávací algoritmus

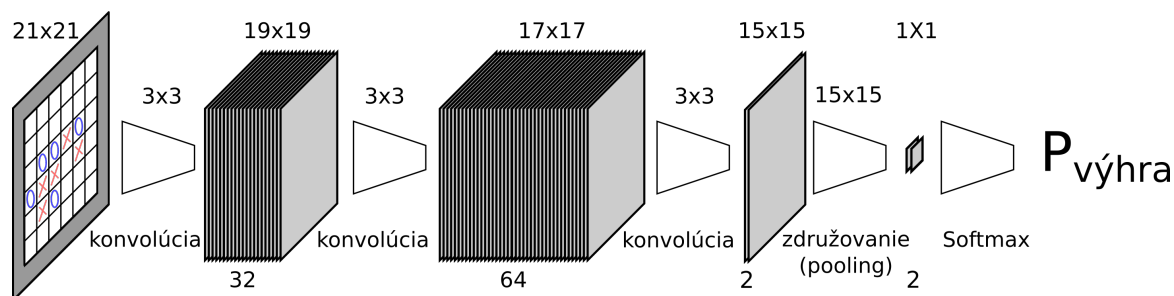
Pri vyhľadávaní sú použité dva algoritmy, ktoré sa inšpirujú algoritmi používanými v AlphaGo. Oba začínajú tým, že vezmú prvotnú konfiguráciu hracej plochy a pomocou strategickej siete vygenerujú prvotný ťah. V prípade, že si to predstavíme ako strom, konfigurácie hracej plochy predstavujú uzly a jednotlivé ťahy, z ktorých vznikajú ďalšie konfigurácie, predstavujú hrany.

Prvá metóda využíva iba strategickú sieť. Z vybraného počtu prvotných ťahov prevedie simuláciu až do ukončenia všetkých simulovaných hier. Z každého vybraného prvotného ťahu vytvorí novú konfiguráciu

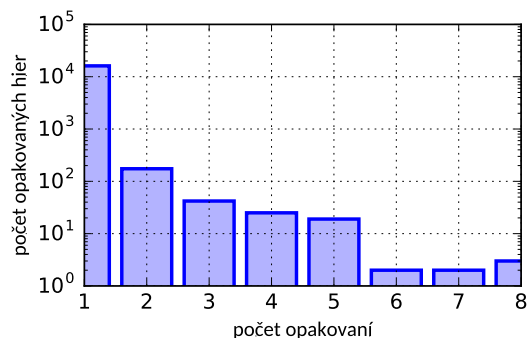
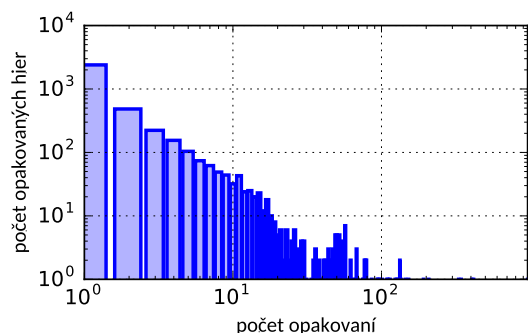
ci nastavením hodnoty políčka podľa súradníc ťahu. Vytvorenú konfiguráciu nakopíruje toľko krát, aký je parameter šírky vyhľadávania. Konfigurácia je ďalej vstupom strategickej siete, ktorá ďalej generuje ťahy a vybraný ťah aplikuje na príslušnú hraciu plochu. Každý ťah je vybraný s pravdepodobnosťou určenou strategickou sieťou. Simulácia končí, keď je na všetkých konfiguráciách splnená podmienka ukončenia hry. Výsledné hodnotenie prvotného ťahu je pomer výhier a prehier. Druhá metóda sa od predošlej líši tým, že simulácia prebieha do určenej hĺbky a aproximácia hodnotenia je určená pomocou ohodnocovacej siete. Narozdiel od AlphaGo, tieto metódy nepoužívajú zľavový parameter, ktorý zvyšuje kvalitu hier ukončených skôr.

2.3 Architektúry sietí

Strategická sieť z obrázku 2, ktorá vyberá ťahy podľa pravdepodobností, má päť rôznych architektur, ktoré majú od dvoch po šesť konvolučných vrstiev. Každá z nich, okrem poslednej, je nasledovaná ReLU vrstvou a vrstvou pre normalizáciu dávky (batch normalization). Pretože zároveň je vstupom Softmax vrstvy [3]. Strategické siete sú navrhnuté tak, aby okolie, o ktorom mala informácia každá pozícia, boli minimálne dve políčka z každej strany. Toto je nutné, pretože kľúčovou



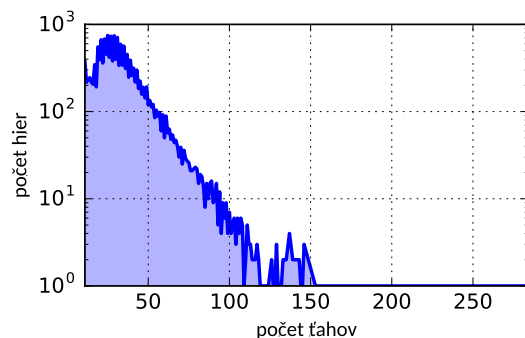
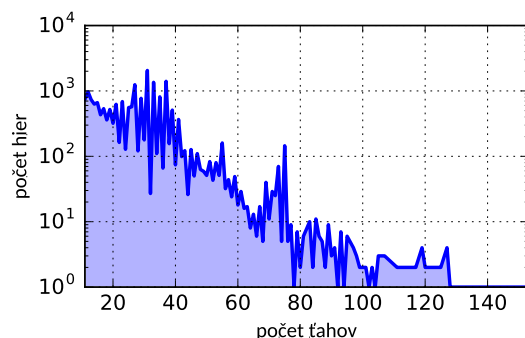
Obrázok 3. Architektúra ohodnocovacej siete s tromi konvolučnými vrstvami. Vstupná hracia plocha je ohraničená výplňou pre označenie okraja plochy. Výstupom sú dve hodnoty určujúce pravdepodobnosť výhry.



Obrázok 4. Histogram hier a ich opakovaní v starej (hore) a novej (dole) vygenerovanej dátovej sade. Pri generovaní novej dátovej sady boli začiatky hier náhodné, čo zabezpečilo minimalizáciu opakovania jednotlivých hier.

úlohou je vyhľadávanie päťice kameňov v rade. Ideálne je, aby okolie bolo ešte väčšie, pretože aj políčka v okolí plochy so stranou o veľkosti päť políčok môžu rozhodovať o kvalite danej konfigurácie.

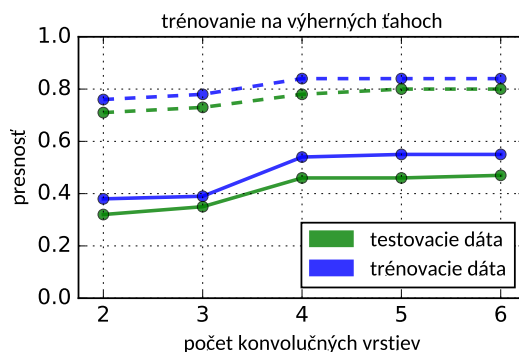
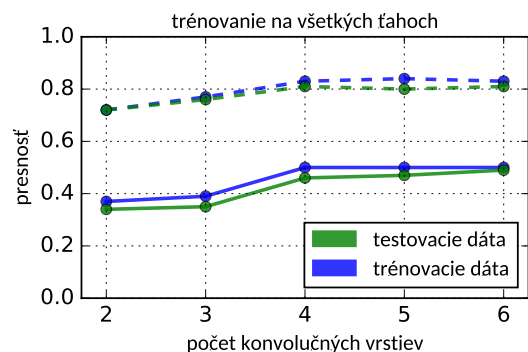
Strategické siete, ktoré sa líšia počtom konvolučných vrstiev, boli navrhnuté tak, aby jedna iterácia trvala približne 30 až 50 milisekúnd. Ich architektúry sú podrobne popísané v tabuľke 1. Čas bol zmeraný pri počítaní na grafickej karte a veľkosti dávky o 512 hracích plochách. Približne rovnaká dĺžka iterácie bola dosiahnutá zväčšovaním a zmenšovaním počtu



Obrázok 5. Histogram hier a ich dĺžky v starej (hore) a novej (dole) vygenerovanej dátovej sade. Najviac vygenerovaných hier malo v oboch dátových sadách dĺžku približne 30 ťahov.

výstupov jednotlivých konvolučných vrstiev.

Zatiaľ nie je zistené, či je pre ohodnocovaciu sieť vhodné sledovať okolie 7x7, alebo väčšie, pretože doteraz navrhnuté neurónové siete neurčovali zmysuplné pravdepodobnosti výhry. Celková architektúra je podobná ako pri ohodnocovacej sieti. Rozdiel medzi strategickou a ohodnocovacou sieťou je v použití združovacej (pooling) vrstvy po každej druhej konvulúcii. To umožňuje zväčšiť sledované okolie, a hodnotenie tým zahrnúť viacero políčok. Celkovo je použitých šesť konvolučných vrstiev nasledovaných Softmax vrstvou. Na obrázku 3 je zobrazená ohodnocovacia sieť s tromi



Obrázok 6. Úspešnosť jednotlivých strategických sietí na testovacej a trénovacej dátovej sade. Plná čiara reprezentuje uhádnutie ťahu a prerušovaná, že hľadaný ťah bol medzi piatimi najpravdepodobnejšími. Kvôli relatívne malej trénovacej dátovej sade sú niektoré neurónové siete mierne pretrénované.

konvolučnými vrstvami.

3. Experimenty

Pre prácu s neurónovými sieťami je použitá knižnica `caffe` a jej nadstavba v jazyku Python. Vďaka tomu bolo možné vytvoriť jednoduché skripty pre tréning a tak tiež vyhodnocovanie sietí pri samotnom hraní. Pre porovnanie schopností umelej inteligencie s dostupnými programami je vhodný herný klient [Piskvork](#). S ním je možné používať programy pre hranie gomoku, ktoré sú zverejnené na rovnakej stránke. Všetky programy sú zoradené podľa skóre, ktoré určuje ich schopnosti. Niektoré majú taktiež zverejnený zdrojový kód.

Dáta pre tréning neurónových sietí boli získané ukladáním hier hraných medzi stiahnutými programami. Vybrané programy boli umiestnené v rebríčku na prvom až dvadsiatom mieste. Pri generovaní dátovej sady boli súperiace programy rôzne kombinované kvôli zvýšeniu variability hier, avšak to nestačilo. Pôvodne bolo vygenerovaných 19 753 hier a iba 3 953 bolo unikátnych. Problém bol, že všetky hry začínali na prázdnej hracej doske, a preto vzniklo málo unikátnych hier. Situáciu vyriešila možnosť začať hru z uloženej konfigurácie a náhodne ju otáčať. To umožnilo zaznamenať 16 831 hier, z ktorých bolo 16 379 unikátnych. Porovnanie dĺžky a opakovaní jednotlivých hier pôvodnej a novej dátovej sady je popísané na obrázkoch 4 a 5.

Z experimentov, zaznamenaných v tabuľke 2, vyplynulo, že strategická sieť dokáže konkurovať nižšie umiestneným programom zo súťaže Gomocup, ale proti vyššie umiestneným už nie je schopná vyhrať. Pozorovaním priebehu hier sa zistilo, že strategická sieť hrá zo začiatku defenzívne, ale má problém zakončiť hru. To môže byť ovplyvnené pretrénovaním vzhľadom na začiatkové pozície.

Pre ohodnocovaciu sieť zatiaľ nebola nájdená vhodná architektúra, preto nebolo možné zaznamenať žiadne experimenty. Doteraz natrénované architektúry neurčovali zmysluplné pravdepodobnosti výhry.

4. Záver

Táto práca sa zaoberala vytvorením umelej inteligencie, ktorá dokáže konkurovať programom, ktoré používajú predom definované pravidlá a heuristiky pre určenie ťahu. Pre tréning je použitá sada hier vytvorená hraním programov zúčastnených v súťaži Gomocup. Týmto postupom sa podarilo vytvoriť tréningovú a testovaciu sadu s približne 20 000 unikátnymi hrami. Natrénovaná strategická sieť dokázala poraziť nižšie umiestnené programy v súťaži Gomocup.

Vytvorená umelá inteligencia môže byť stále zlepšovaná. Je možné zdokonaľiť prehľadávanie stavového priestoru, prípadne experimentovať s väčšími architektúrami strategickú a aj ohodnocovacú sieť. Taktiež sa dá experimentovať s parametrami jednotlivých vrstiev, prípadne s parametrami tréningu a sledovať vplyv na výsledok tréningu.

Podakovanie

Chcel by som poďakovať vedúcemu mojej bakalárskej práce, Ing. Michalovi Hradišovi, Ph.D. za pomoc a odborný dohľad. Taktiež by som chcel poďakovať za prístup ku výpočtovým a úložným zariadeniam patriacim skupinám, ktoré prispievajú k Národnej infraštruktúrnej sieti MetaCentrum poskytovanej pod programom "Projekty veľkého výskumu, vývoja a inovácii infraštruktúr" (CESNET LM2015042).

Literatúra

- [1] David. Silver et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, Jan 2016. Article.
- [2] Yuxi Li. Deep reinforcement learning: An overview. *CoRR*, abs/1701.07274, 2017.
- [3] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-Margin Softmax Loss for Convolutional Neural Networks. *ArXiv e-prints*, December 2016.