REINFORCEMENT LEARNING

Ladislav Martínek
martilad@fit.cvut.cz

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra aplikované matematiky

12. listopadu 2019

ABSTRACT

V této práci je posáno jedno ze tří paradigmat strojového učení a to reinforcement learning – zpětnovazební učení. Dalšími jsou: učení s a bez učitele. Nejprve je posána samotná metoda a specifikovány její hlavní časti, které jsou klíčové pro simulaci na počítačích. Dále je popsán kompromis mezi exploitatací a explorací a nejčastěji používané metody v tomto přístupu k učení. Ke konci práce jsou popsány příklady, na kterých je zpětnovazební učení využito jako hlavní princip. Ve všech částech jsou metody dávány do souvislostí s učením v reálném světě a jak je z něj přístup ve strojovém učení inspirován.

Klíčová slova reinforcement learning · zpětnovazební učení · machine learning · strojové učení · posilové učení · umělá inteligence

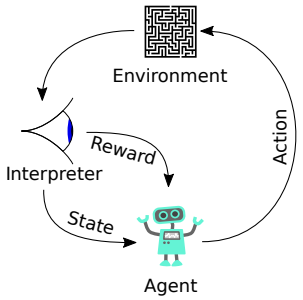
1 Úvod

Když se zkusíme zamyslet nad tím, jak jsme se odmalička učili, a jak učení probíhalo, tak nás jako první metoda napadne právě reinforcement learning (dále budu používat zkratku RL), neboli také učení na základě zpětné vazby. Určitě jsme učení takto přímo nenazývali, ale zpětné vazby dostáváme odmalička. Zpětnou vazbou může být například: pocit chladu pokud se dotkneme sněhu. Takovou vazbu máme přímo od prostředí. Další zpětnou vazbu můžeme dostávat například od rodičů a určitou zpětnou vazbu od prostředí dostáváme téměř neustále. Na tomto způsobu je založeno i jedno z hlavních paradigmat strojového učení. V tomto případě není systému poskytováno žádné mapovaní správných odpovědí na daný vstup nebo interakci. Učený subjekt musí vyzkoušet, co mu přinese nějvětší užitek (největší odměnu). V této práci popíši RL ve strojovém učení i s problémy, se kterými potýká, a podobnosti a odlišnosti proti tomuto učení v realném světě.

2 Reinforcment learning

Reinforcment learning, v češtině také zpětnovazebné učení nebo posilové učení, je způsob učení na základě odezvy a zpětné vazby z prostředí, ve které se učený subjekt pohybuje.

Teorie RL poskytuje normativní pohled, který je hluboce zakořeněn v psychologickém a neurovědním pohledu na chování zvířat, jak mohou subjekty optimalizovat svoje chování a kontrolovat prostředí [6].



Obrázek 1: Clasický scénář RL. Agent prování akce v prostředí, které je nějakým způsobem interpretováno. Z interpretace je odvozena odměna pro agenta a reprezentace stavu prostředí, který má agent k dispozici. [5]

V rámci strojového učení se jedná přímo o jedno z paradigmat učení, které je v poslední době velice populární současně s tím, jak roste výkon počítačů a je možné simulovat složitější agenty a prostředí. Na stejné úrovni rozdělení jako RL stojí učení s učitelem (angl. supervised learning) a učení bez učitele (angl. unsupervised learning). Oproti učení s učitelem (jeden z nejčastějších přístupů ve strojovém učení) není v RL poskytnut popis jednotlivých situací, ke kterým je by byla přesně určena ta správná akce. Rozdíl oproti učení bez učitelel je ten, že učení bez učitele slouží většinou k nalezení struktury v neoznačených datech [8].

V RL zpětnou vazbu nejčastěji představují odměny nebo tresty. Učení je možné buď s obojím nebo například jen s odměnami nebo tresty. Učený subjekt se snaží o maximalizaci takové odměny nebo minimalizaci trestů.

Podle [8] učený agent musí být schopen vnímat stav svého prostředí a musí být schopen provádět akce, které toto prostředí mění. Celkově je také agent řízen svými cíly, kterých se snaží dosáhnout a může k tomu využít různých cest, které můžou vytvořit různé cesty pro různé agenty.

Ze všech forem strojového učení (dále jen ML) je RL nejblíže tomu, jak se reálně učí lidé a ostatní zvířata. Mnoho postupů v RL bylo původně inspirováno právě biologický systémem učení [8]. I ostatní části ML lze nalézt v biologickém systému učení, ale ty jsou především tom našel lidském, kde můžeme mít například učení s učitelem. Učíme z příkladů, kde známe správné výsledky. Toto jsou pouze malé části, protože za nějakou takto naučenou látku jsme ve škole byly odměněni známkou nebo body. Dostali jsme zpětnou vazbu, jestli způsob jakým jsme se učili byl správný nebo věnovaný čas dostatečný, či nikoliv.

3 Reprezentace a simulace RL

Když se podíváme na RL v našem životě, tak lze hodně jednoduše zobrazit pomocí obrázku 1. Většinu částí jsem již zmínil v části 2. Ale v této části popíši jednotlivé součásti RL a jejich rozdíly mezi reálným světem a jejich simulací v počítači. Bude se jednat o agenta a prostředí, v rámci kterých popíši interpretace stavu prostředí a odměny pro agenta. Dále agent musí mít nějaké cíle, kterých se snaží dosáhnout. Tyto cíle musí být staženy vždy na daný stav prostředí, ve kterém se nachází. Ty jsou nutné proto, aby agent měl nějaký směr (volbu akcí), kterým se vydat.

3.1 Prostředí

Ve strojovém učení je v [7] jako první zmíněna funkce generující odměnu pro agenta (nutná součást prostředí), protože každý agent je na základě svých akcích odměnován. V ML je nejčastěji odměnou celé číslo, které se spjato s nějakým stavem a akcí. Stavem je popsáno aktuální prostředí, které muže daný agent pozorovat např. rozehraná šachová partie. Jak je vidět na obrázku 1, stav je nějak interpretován a to nejčastěji nějakými senzory agenta.

Odměna pro agenta je vždy svázána se stavem a konkrétní akcí. Tento koncept stavů a akcí je velice obecný, ale v ML používaný. Akce je agentovo jakékoliv rozhodnutí, které agent může vykonat s stav je jakýkoliv fakt, který může vzít agent v úvahu při rozhodování [7].

V ML je ohodnocení poměrně omezené oproti reálnému světu. Zpětnou vazbu můžeme rozdělit například na objektivní a subjektivní zpětnou vazbu. Při objektivní zpětné vazbě nevznikají pochyby a v případě ML je zpětná vazba vždy objektivní – určena funkcí. Například tabulka s jasně danými odměnami. Subjektivní zpětná vazba může vyvolat pochyby o její důležitosti a pravosti, muže jít pouze špatné rozhodnutí o dané odměně. V biologickém světě je mnohem častější. Jako příklad mě napadají například odměny v práci, které můžou představovat objektivní zpětnou vazbu,

kdy může být odměna závislá na počtu vyrobených kusů. Nebo subjektivní, kdy rozhoduje například aktuální nálada manažera nad vámi. Odměna pak může mít i opačný efekt než podpoření učení.

Při výcviku psa by pochvala jako pozitivní odměna příliž nefungovala, ale na nějaký pamlsek už reaguje většina psů a jsou poměrně rychle schopni provádět požadované akce. U lidí je množina možných odměn ještě rozsáhlejší a může být představována i slovní pochvalou, finanční odměnou a mnoho dalšími. Vytvořit funkci v ML, která by toto zahrnovala je velice optížné až nemožné. Prostředí s odměnami je omezováno (číslo) a většinou vždy zamířeno na pár konkrétních cílů, kterých se má učením dosáhnout. Více v kapitole 6.

3.2 Agent

Agent je součástí konkrétního prostředí. Hlavním cílem agenta je mapování možných stavů na možné akce [7]. V ML může být implementováno tabulkou nebo měnícím se prostředím. Jedná se o asociace, které známe i z našeho života.

Agenti mají většinou nějaké senzory, kterými pozorují okolní prostředí. Jako agenta v prostředí, můžeme uvažovat například samořídící auto, které má různé senzory, pomocí kterých se orientuje v prostoru (lidar, radar, kamery , atd. . .). Pro takového agenta je možné použít RL na jeho učení. Většinou ale není použit pouze RL, ale i další metody učení, tak že prostor akcí je striktně velmi omezen, protože provoz se svázán zákony. Vazbu z prostředí získavájí i zvířata a lidé. K získání stavu prostředí nám slouží oči, hmat, sluch, ale například i čich a další. . .

Informace ze senzorů můžeme nazvat vnitřními nebo také vlastními. Na druhé straně jsou vnější, které můžeme získat od ostatních agentů jako ucelenou informaci. I s příkladem jsou více popsány v 6.2.

4 Exploitation vs. Exploration

Jak se píše v [4], jeden z hlavních rozdílů mezi učením s učitelem a RL je, že v případě RL musí agent explicitně prozkoumávat prostředí akcemi. Potom je zásadní, jaký udělá kompromis. Zdali bude prostředí velmi aktivně prozkoumávat (exprorace), ale za cenu mnohých neúspěchů nebo v případě, že nalezne akce, ze kterých plyne nějaký profit se začne zaměřovat na tyto akce a provádět tzv. exploitaci. Může existovat mnoho různých akcí, které by mohli být optimálnější, ale pokud bude prozkoumávat pouze okolí těchto, tak na ně nikdy neodsáhne. Uvízne v tzv. lokálním maximu.

Když se podíváme do reálného světa můžeme vidět, že i tyto dva přístupy jsou zde vidět. Je to pozorovatelné jak mezi zvířaty tak mezi lidmi. Existují takový jedinci, kteří zkoušejí vše, co se jim nabídne nebo se snaží co nejvíce prozkoumat možnosti a hranice. Přirovnání k vedoucím manažerským pozicím to může být například způsob vedení firmy. Vysoká explorace může zapříčinit vysoký zisk, ale také je stejná možnost rychlého pádu při radikálních změnách směřování firmy a hledání optimální cesty. Na druhou stranu exploitace může pomoci upevnit pozici na konkrétním segmentu a stavu, ale růst firmy například již nebude tak rychlý, jakýho my se mohlo třeba dosáhnout vysokou explorací.

Agenti v ML musí volit nejen takové akce, které optimalizují zisk, ale také akce, které dovolí dostatečné prozkoumání prostoru a dovolí se naučit nové akce. Každá akce musí být několikrát vyzkoušena, aby se zíkal statictický odhad její očekávané odměny [8]. V ML jsou metody využity především k řešení úloh, kde je žádoucí dosáhnout co nejlepšího výsledku, narozdíl od reálného světa, kde nemusí být ve výsledku takový problém uváznutí v lokálním maximu. Protože z pozice osoby (agenta) můžou být také akce optimální. Osoba je například spokojená, záleží pouze na cíli, který osoba má a může mít mnoho podob. Pro potřeby ML, kde je cíl a učel známý, nám tedy explorace může pomoc se z takového lokálnímo maxima dostat a příblížit se optimu nebo ho přímo dosáhnout.

5 Metody RL

V této kapitole a především jejích podkapitolách popíši některé metody, které jsou v rámci RL používané. Tyto metody lze podle knihy [8] rozdělit na metody, kde akcí a stavů není tolik a je možné je reprezentovat tabulkou. Tyto metody jsou schopny většinou nalézt přesné řešení v dané množině akcí a stavů. Na druhé sraně jsou aproximační metody, které většinou najdou pouze přibližné řešení, ale za to je možné je použít na mnohem větší problémy.

5.1 Dynamické programování

Dynamické programování je první z method pro řešení RL. Problém je formulován jako Markovův rozhodovací proces, na kterém je možné dynamické programování použít. Markovův proces se skladá z mnnožiny stavů, akcí, pravděpodobností, že ve stavu s bude provedena akce a a okamžitý užitek po přechodu do nového stavu pro provedení konkrétní akce na současném stavu [8].

Dynamické programování je popsané v [8] a předpokladáme pro něj konečnou množinu stavů a akcí. V opačném případě je problém reprezentace v konečné paměti. Dynamické programování je založeno na principu, že podstrategie optimální strategie je opět optimální. Toto není použitelné na jakoukoliv úlohu, ale pro ty, kde platí, lze využít dynamické programování.

V dynamickém programování jsme schopni výsledek většího podproblému vyjadřit pomocí menších problému, které jsme už jednou vyřešili, není tedy nutné některé jednoduché úlohy řešit opakovaně. V tomto je přímo vidět učení z normálního světa. Příklad mě napadá třeba, když dáme dítěti šroubovák a nějaké šroubky a prostor. Za chvíli se naučí jak se šroubovákem pracovat a tuto podúlohu poté použije v jakémkoliv věku. Třeba když si bude stavět sřínku – nebude se opakovaně učit šroubovat. Přesně tohodle principu je využito i při dynamickém programování, kde se výhradně pracuje s tabulkou, která je na základě stavů daného prostředí a možných akcí vypňována, až je dosaženo požadováného cíle.

5.2 Monte carlo metody

Monte carlo metody jsou stochastické metody, které jsou hojně využívané tam, kde není možné prozkoumávat celý stavový prostor souvisle. Narozdíl od přechozích části, zde nepředpokládáme znalost všech stavů a akcí, celkovou znalost prostředí. Monte carlo metody vyžadují pouze zkušenost, nějaký vzorek sekvencí stavů, akcí a odměn z aktuální interakce s prostředím [8]. Učení ze zkušenosti je velice překvapující, protože nevyžaduje znalost dynamiky prostředí a přesto může dosáhnout dobrého řešení. I zde potřebujeme znát model, ale potřebujeme ho pouze ke generování ukázkových přechodů a nikoliv k rozdělení pravděpodobnosti do tabulky všech možných přechodů, jako tomu bylo u dynamického programování.

Je přímo využito průměrování odměn za danou akci v daném prostředí. Model se učí z odměn, které dostává. V monte carlo metodách je používána určitá složka náhody, která slouží k výběru akcí a prozkoumání prostoru s tím, že je upravována podle zkušenosti jaké bylo dosaženo. Akce bez odměny mohou být například jedno zopakovány a vyzkoušeny, ale průměrováním budou velmi rychle vyloučeny. Pokud učíme psa, tak ho pomocí zkušeností naučíme, že třeba nesmí vyhrabávat koš, kde za takovou akci přijde ještě trest, tak již takové chování nikdy nezopakuje, stejně tak se bude chovat agent v případě učení pomocí monte carlo metod. Více je vidět v kapitole 6.

5.3 Temporal difference (TD) learning

TD je kombinací metod monte carlo učení a dynamického programování. Stejně jako v monte carlo metodách se může učit přímo ze zkušeností, ale také je aktualizována tabulka částečných naučených akcích jako v dynamickém programování [8]. Algoritmy se mohou i navzájem prolínat a využívá se vždy silných stránek daného algoritmu. TD upravují a vybírají akce podle toho jaký bude mít výsledek do budoucnosti. Ještě dříve než je znám konečný výsledek, jsou zváženy možné akce a jejich výsledky. Monte carlo metody naopak rozhodují až podle výsledku, kterého dosáhly a upraví pravděpodobnosti a průměry.

5.4 Aproximační metody RL

Tyto metody jsou podrobněji rozepsány v [8]. Zde jen stručně popíši, co metody představují. Jsou rozšířením předchozích metod pro práci s libovolně velkými prostory stavů a akcí. V takovém prostoru nemůžeme očekávat, že najdeme optimální řešení pro náš řešený problém, ale můžeme se mu alspoň příblížit s omezenými výpočetními prostředky. Problém představuje jednak uložení tak velké tabulky do paměti, ale také jak takové tabulky vyplnit v konečném čase. V takto složitém prostředí zřejmě nikdy neuvidíme dvakrát úplně stejný stav. Musíme například umět najít podobnosti mezi stavy. Klíčová je v tomto učení generalizace neboli zobecnění. Jednotlivé zobecňující funkce můžou být již například výsledkem nějakého učení s učitelem a my je využijeme (detekce objektu na obrázku). Dále se systém musí naučit reagovat na jednotlivé objekty pokud jsou na obrázku. Tímto způsobem se snaží postupně učit reagovat na podměty, ale ne na úplně každý, ale zobecňovat své reakce (třeba pokud uvídí kočku) pro maximalizaci odměny.

6 Příklady RL

V této části uvedu příklady, na které jsem narazil a je na nich vidět, kde se všude může RL uplatnit. Na některé jsem narazil již dříve v rámci ML. První se podívám na učení pomocí RL, kde je simulován člověk při ovládání hry. Druhý příklad nepřichází z oboru ML, ale učení zvířat, konkrétně opic, kde je skloubeno učení pomocí zpětné vazby v kolektivu a je použito na posledním příkladu, kde je učení chování postaviček.

6.1 Human-level control

V článku [6] popisují vytvoření a trénování modelu pomocí RL. Tento příklad se liší od mnoha příkladů na RL, protože agent hry ovládá přes stejné rozhraní jako člověk a s ním je také srovnáván v různých hrách. V článku se jedná o hry Atari 2600 a stejné rozhraní.

Cílem bylo vytvořit agenta, který by si uměl poradit s mnoha různými úkoly v různých hrách. K učení je využito RL a také konvoluční neuronová síť, která se využívá především pro zpracování obrazových dat. Výstupem sítě jsou ovládací prvky joisticku. Učení probíhalo pomocí zpětné vazby, kterou bylo pouze dané skoré a to bylo využito k úpravě vah a přizpůsobení algoritmu.

Jako vstup byla použita kamera s rozlišením 210 x 160 pixelů. Algoritmus byl tedy schopný se učit z nepříliš kvalitních obrazových dat a scóre jako zpětné vazby. V žádné hře nebyly použity žádné jiné informace.

V článku jsou prezentovány výsledky v procentech jako porovnání. 0 % je pro náhodně volené akce a 100 % pro lidského hráče. Není jasné jakých kvalit dosahoval lidký hráč, ale i přes to je algoritmu podařilo přibližně v polovině her porazit lidského hráče. Nějlepších výsledků dosáhnul ve hře PinBall, kde byl na 2539 % skóre oproti hráči. Velice rychle se agent naučil nějakou optimální taktiku. Rychleji se učil hry, kdy nebylo potřeba tak velké koordinace a využívání více ovládacích prvků.

Také nebyla zmíněna délka učení pro takové výsledky, ale bylo by zajímavé srovnat rychlost učení dítěte a algoritmu, jak by se vyvíjeli jejich skóre a učili se ovládat konzoli a hledali optimální cesty v jednotlivých hrách. V článků bylo ukázáno, že RL je možné dobře využít při učení sofwarového agenta a jeden agent se dokázal jen pomocí skóre naučit hrát různé jednoduché hry a zlepšovat se v nich.

6.2 Monkeys

Experiment byl poprvé zmíněn v [3]. Není vůbec jasné, zdali byl experiment proveden a nebo ne, ale to není tak důležité, je zajímavé jak se opice učili, kde na začátku to byla právě zpětná vazba na akci, na které je RL založeno.

Při experimentu bylo umístěno 5 opic do místnosti, kde byl žebřík pomocí kterého se dalo dosáhnout na banány. Samozřejmě na žáčátku se první opice vydala pro banány, ale všechny byly skropeny studenou vodou, která představovala zpětnou vazbu, že na žebřík nesmí. Zanedlouho se to naučili, že pro banány se nesmí. V tento moment byla vyměněna jedna opice. Nová opice se vydala okamžitě k žebříku, ale ostatní jí okamžitě zadržely a stáhly, tak se naučila, že na žebřík pro banány se nesmí. Takto byly postupně vyměněny všechny opice, ale tato naučená zkušenost zůstala. Toto chování dokázali udržet i přes to, že již nezbývala žádná opice, která by přímo zažila tu zkušenost. Tímto se učení urychlylo.

Pokud byl experiment pravý, tak je velice zajímavé, jak se dokázala naučená zkušenost uchovat a předat dále i na jedince, kteří nezažili přímo zpětnou vazbu. Toto sdílení může velice urychlit učení. Je vidět na dalším příkladu, kde agenti jsou schopni se naučit týmové práce, aby dosáhli odměny co nejrychleji. Tento příklad se týká přímo ML a RL.

6.3 Open AI

Tento experiment je popsaný na [2] a je k němu i mnohem více technický članek [1] a částečně je také dohledatelná implementace použitých algoritmů. Tento experiment je mnohem novější a byl publikován v září 2019. K přístupu RL z prvního článku, přídává také myšlenky z experimentu druhého. Jsou využity i vnější informace k učení, jak bylo zmíněno už v části 3.2. Jsou to informace od ostatních agentů, kteří jsou s ním v týmu. Agenti tak mohou vytvářet týmové strategie a předat své zkušenosti ostatním agentům.

Je daná mapa, kde jsou nějaké předměty jako rampy, bedny a posuvné stěny. Na mapě se pohybují dva týmy postaviček – červení a modrý. Tito agenti mají určený cíle. Modrý nesmí být vidět napřímo červeným a červený musí vidět modrého a za tyto akce jsou odměněny. Modří mají na začátku čas si mapu připravit, tento čas je také snižován. Zde je k učení využito také pouze RL s tím, že agenti v týmů sdílejí naučené dovednosti.

Agentům nebyla dána žádná informace pro to, že mají s objekty hýbat nebo spolupracovat, pouze cíle zmíněné výše. Odměna byla 1 pokud nebyli modří vidět a -1 pokud je dokazali červení najít a naopak pro červené. Agenti při učení dokázali tvořit různé strategie a vždy využít prostředí.

V článku je zmíneno několik strategií, které se postupně agenti učili, já zde zmíním jednu. Na mapě je místnost se dvěma dveřma, krabicemi, kterými lze dveře blokovat a rampou, pomocí které lze přeskočit stěnu. Postup učení byl takový, že agenti se nejprve naučili zabarykádovat v místnosti, aby na ně červení nemohli.

Tady je krásně vidět, že učení probíhá pouze pokud je nutnost odměny, aby byla dostatečná motivace se učit nové věci. V moment, kdy dostali modří odměnu, když se na ně červení nedostali, tak se neučili nic nového a byly s odměnou spokojeni, ale červení odměnu přestali dostávat a tím bylo posunuto jejich učení. Postupně se naučili použít rampu k tomu, aby přeskočili zeď a odměnu dostali oni.

Podobné chování lze pozorovat i v biologickém světě. Pes vždy bude hledat tu nejjednoduší cestu k odměně a pokud bude odměnu dostávat zadarmo nebude se učit nové věci. Tady je vidět přímá podobnost tohoto učení, kde jsou vidět principy z přírody.

Při iteraci byla snaha modrých o získání odměny, naučili se tedy si rampu schovat. Čím více bylo iterací tím více koordinovali svoje pohyby tak, že každý měl rozdělen svůj úkol tak, aby se optimalizovala rychlost schování. Tato optimalizace pohybu byla učena snižujícím se časem na přípravu.

Učení tedy může být urychleno, pokud se k učení přídá také soutěživost a spolupráce více agentů, odměnu může mít pouze jeden tým a přesně tímto může být učení ještě posíleno.

6.4 Závěr

V práci jsem představil zpětnovazební učení (angl. reinforcement learning). Popsal jsem hlavní principy a popsal metody, které jsou využívány. V jednotlivých bodech jsem hledal souvislosti s biologickým učením, kde je učení pomocí zpětně vazby hlavním učícím mechanismem a odtud také bylo v ML inspirováno. Na závěr jsem představil příklady, které mě nejvíce zaujali.

RL je velice populární metoda ve strojovém učení, ovšem jako všechny ostatní metody není všehoschopná, nehodí se na všechny typy úloh a má vlastní omezení a limity. První je především nutnost specifikovat funkci, která generuje odměnu a může být velice komplexní a tím složitější, čím větší problém budeme chtít řešit. Dále je potřeba přesně specifikovat cíl. Může se stát, že problém bude tak komplexní, že bude problém cíl specifikovat nebo se může v průběhu času měnit na základě jiných skutečnostích. Obecně jsou však tyto metody používané a dosahuje se pomocí nich obdivuhodných výsledků například učení v kapitole 6.3.

Tyto metody jsou více než jiné inspirovány biologickým učením v přírodě. Jakákoliv zpětná vazba nám pomáhá se učit. Narozdíl od počítačového RL, máme výhodu v možnosti přijímat i složitější zpětnou vazbu (prozatím tu výhodu máme). RL je jeden z přístupů v ML, který bude určitě dále rozvíjen (hlavně v možnosti reprezentovat komplexnější systémy) a bude použit jako základ pro vytváření složitějších autonomních systémů, kde může být doplněn dalšími metodami.

Reference

[1] Baker, B.; Kanitscheider, I.; Markov, T.; aj.: Emergent Tool Use From Multi-Agent Autocurricula. 2019, [cit. 2019-11-18], 1909.07528.

[2] Bowen Baker and Ingmar Kanitscheider and Todor Markov and Yi Wu and Glenn Powell and Bob McGrew and Igor Mordatch: Emergent Tool Use from Multi-Agent Interaction. [online], 9 2019, [cit. 2019-11-18]. Dostupné z: https://openai.com/blog/emergent-tool-use/

[3] Hamel, G.; Prahalad, C. K.: *Competing for the Future*. Harvard Business Press, 1996.

[4] Kaelbling, L. P.; Littman, M. L.; Moore, A. W.: Reinforcement learning: A survey. *Journal of artificial intelligence research*, ročník 4, 1996: s. 237–285, [cit. 2019-11-15].

[5] Megajuice: Reinforcement learning diagram.svg. 2017, [cit. 2019-11-15]. Dostupné z: https://commons.wikimedia.org/wiki/File:Reinforcement_learning_diagram.svg

[6] Mnih, V.; Kavukcuoglu, K.; Silver, D.; aj.: Human-level control through deep reinforcement learning. *Nature*, ročník 518, č. 7540, 2015: str. 529, [cit. 2019-11-13].

[7] Sutton, R. S.; Barto, A. G.: Reinforcement learning. *Journal of Cognitive Neuroscience*, ročník 11, č. 1, 1999: s. 126–134, [cit. 2019-11-13].

[8] Sutton, R. S.; Barto, A. G.; aj.: *Introduction to reinforcement learning*, ročník 2. MIT press Cambridge, 1998, ISBN 978-0-262-19398-6, [cit. 2019-11-13].