Effect of unsupervised pretraining in deep reinforcement learning

Michal Chovanec, michal.nand@gmail.com
Peter Šarafín, peter.sarafin@gmail.com

Keywords:reinforcement learning, function approximation, SARSA, Q-learning, neural network, unsupervised training
Abstract:

Článok sa zaoberá vplyvom unsupervised learning na celkový priebeh reinforcement learning tréningu. Ako aproximátor funkcie ohodnotení je zvolená hlboká neurónová sieť. Overovali sme hypotézu, že unsupervised predtrénovaná sieť sa v režime supervised učí rýchlejšie. Zároveň sa ale očakáva, že obe siete po dostatočnom počte iterácií učenia poskytnú rovnaké výsledky. Dalšia testovaná hypotéza je vplyv riedkosti váh na kvalitu výsledku - dnešné technické prostriedky sú navrhnuté na výpočty hustých sieti **TODO citovat**, v budúcnosti, by však hárdverovo akcelerované riedke siete mohli priniest výrazný nárast výkonu **TODO citovat**. Dôležité je preto overiť vplyv riedkych váh na výsledok.

1 Introduction

Reinforcement learning umožuje tránovať agenta v Markovovich rozhodovacích procesov tak aby maximalizoval dosiahnutú odmenu. Tradične sú používané algoritmy Q-learning **TODO citovat** a SARSA **TODO citovat**. V tomto článku je použítý algoritmus SARSA, popísaný ako

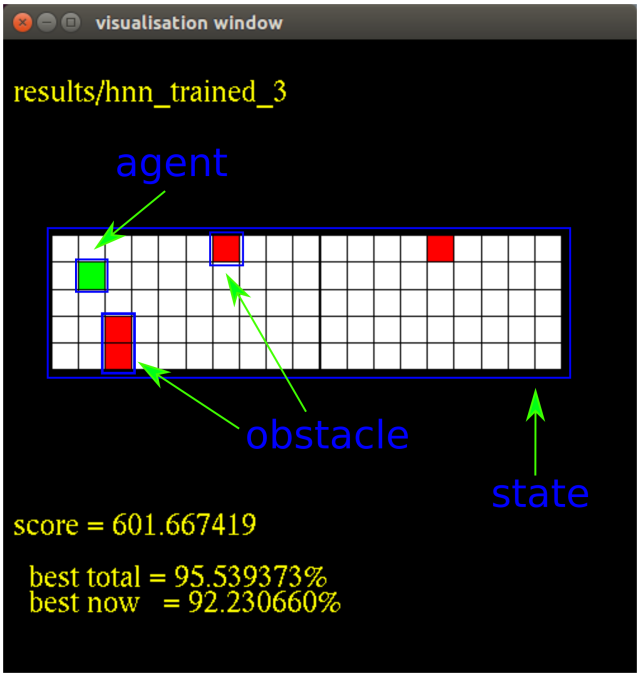
$$Q'(s,a) = (1-\alpha)Q(s,a) + \alpha\Big(R(s,a) + \gamma Q(s',a')\Big) \tag{1}$$

kde
$Q(s,a)$ je ohodnotenie akcie a v stave s,
$R(s,a)$ je odmena za vykonanie akcie a v stave s,
$\alpha \in (0,1)$ je rýchlosť učenia,
$\gamma \in \langle 0,1\rangle$ je discount factor,
a $Q'(s,a)$ je nová hodnota ohodnotenia.

Pre malé stavové priestory (väčšinou len učebnicové príklady) je možné použiť tabuľku na uloženie Q hodnôt. Pre veľké stavové priestory - je nevyhnutná aproximácia Q hodnôt. Často používané sú lineárna kombinácia bázových funkcií **TODO citovat** alebo dopredná neurónová sieť **TODO citovat**.
Rekurentná povaha rovnice 1 spôsobuje že učiaci proces je neefektívny a pomalý.
Naša myšlienka je zameraná na unsupervised predtrénovanie siete a až po tom trénovať použitím SARSA algoritmu.

2 Popis experimentu

Agent bol umiestnený do hry znázornenej na obrázku 1. Úlohou je preskakovať a podliezať červené prekážky, agent má teda na výber dve akcie. Prekážky sa generujú náhodne, a postup je sprava do ľava. Za úspešné vyhnutie sa kolízií získa agent odmenu $+0.3$, za neúspech -1.0. Toto je nevyváženie je dôležité, pre kvalitatívne overenie algoritmu, aby náhodnými akciami nikdy nedosiahol kladné skóre.
Stav sú jednotlivé políčka hry, v tomto prípade 5x19 (95 prvkový vektor). Hodnota 0 znamená voľné (biela), hodnota 1 reprezentuje agenta (zelená), a hodnota -1 je zvolená pre prekážku (červená).

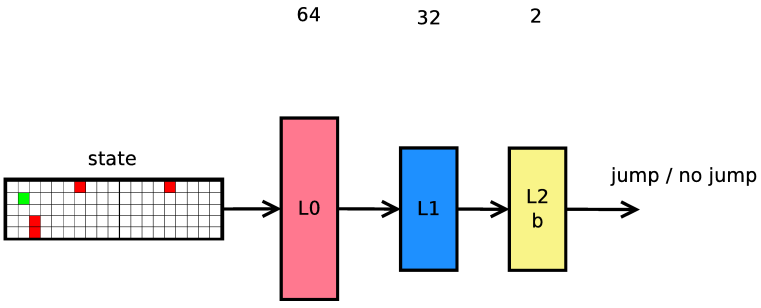


Obr. 1: Testing arcade game

V hre sa sleduje niekoľko veličín. Celkové skóre je dané súčtom jednotlivých odmien. Ďalej sme zaviedli ďalšie dve pomocné veličiny - best_total a best_now.
Veličina best_total ukazuje, koľko percent času volil agent správnu akciu - vyhol sa kolízií, od začiatku behu hry [1].
Veličina best_now ukazuje ukazuje voľbu najlepšej akcie len za posledné obdobie - je použitý priemer s exponenciálnym zabúdaním.

2.1 Topológia siete

Pre porovnanie sme zvolili dve siete. Vstupom je stav hry 5x19 (95 prvkový vektor), výstupom je hodnota $Q(s)$ pre každú akciu - skočit / neskočiť.
Prvá siet je dopredná sieť (FNN) s dvoma skrytými vrstvami, topológia je na obrázku 2. Čísla nad vrstvami sú počty neurónov. Skryté vrstvy majú aktivačnú funkciu ReLU, výstupná vrstva lineárnu aktivačnú funkciu.

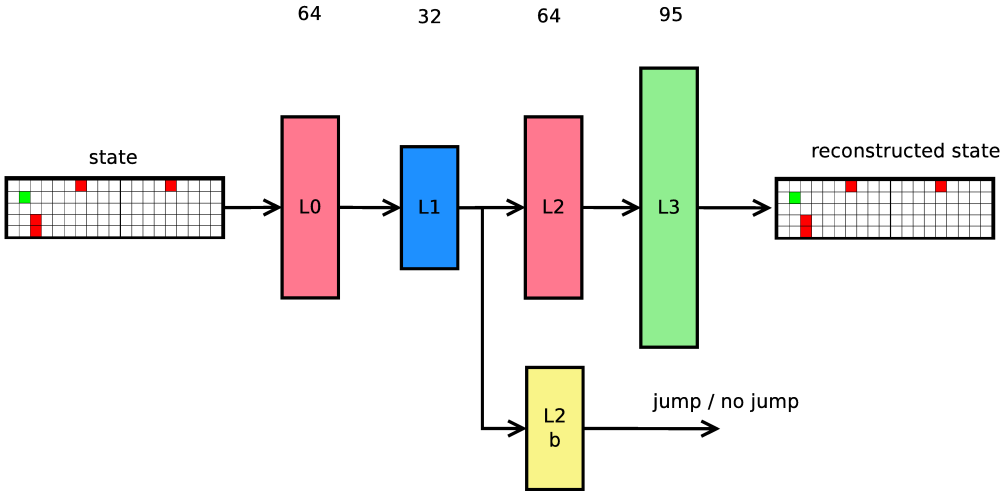


Obr. 2: Supervised network architecture (FNN)

Druhá sieť je tvorená hlbokým autoenkóderom a doprednou sieťou (AE + FNN). Skryté vrstvy majú aktivačnú funkciu ReLU, výstupná vrstva lineárnu aktivačnú funkciu. Jej topológia je na obrázku 3. Vo fáze unsupervised sa trénuje autoenkóder - z 95 prvkového vstupu, postupne extrahuje príznaky, až na počet 32. Následne ďalšie dve vrstvy urobia rekonštrukciu vstupu. Ako učenie sme zovlili metódu stacked autoencoder **TODO citovat**:

[1] Orientačne sme zistili, že ľudský hráč je schopný hrať v intervale od 91% do 93%.

najskôr sa urobí bypass medzi L1 a L2 vrstvami a trénujú sa len vrstvy L0 a L3, po zvolenom počte itererácií sa pripoja vrstvy L1 a L2 a autoenkóder sa dotrénuje.
Po natrénovaní autoenkódera sa odpoja vrstvy L2 a L3. Pripojením vrstvy L2b sa sieť už trénuje supervised, podľa rovnice 1.



Obr. 3: Unsupervised + Supervised network architecture (AE+FNN)

2.2 Parametre experimentu

Experiment má mnoho parametrov, a bolo potrebné ich zafixovať - dnes nie je možné analyticky stanoviť ich optimálne hodnoty. Hyperparametre sietí boli experimentálne zvolené tak aby nedochádzalo k divergencií váh. Úmerne tomu boli upravené počty iterácií učenia.
Veľkosť batch bol zvolený na 1000 po sebe idúcich stavov. Každý batch mal pri tréningu 10 behov (epoch). Parameter SARSA algoritmu γ bol zvolený 0.95, parameter α je pri použití neurónovej siete ukrytý v learning rate parametry siete. Zvolená bola ϵ greedy stratégia voľby akcií. Voľba horšej akcie bola počas tréningu mala prevdepodobnosť 0.1, počas testovania 0.0 - vyberala sa len najlepšia akcia.
Siete boli učené algoritmom backpropagation. Riedkosť váh sa dosahovala pomocou L1 normy. Gradientová metóda učenia váh potom prejde na vzťah

$$\Delta w = \eta E x \frac{df(y)}{dw} - \lambda sgn(w) \tag{2}$$

kde
E je chyba neurónu,
x je vstup do neurónu,
y je výstup neurónu,
f je aktivačná funkcia,
η je learning rate ,
λ je parameter riedkosti váh.

Hyperparametre siete sú zhrnuté v tabuľke 1.

Tabuľka 1: Hyperparametre siete

	FNN sparse	FNN no sparse	AE+FNN sparse	AE+FNN no sparse
unsupervised iterations	0	0	100000	100000
supervised iterations	200000	200000	200000	200000
iterations per slice	0	0	50000	50000
learning rate	0.0005	0.0005	0.0005	0.0005
init weight range	0.1	0.1	0.1	0.1
dropout	0	0	0	0
lambda	0.00000001	0	0.00000001	0

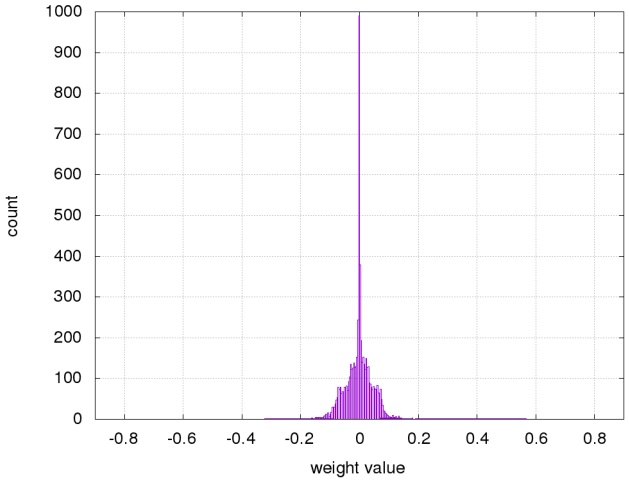
Menili sa parametre **unsupervised iterations**, **iterations per slice** a **lambda**. Parameter iterations per slice je počet iterácií na jednu úroveň stacked autoenkódera.

Sputené boli teda 4 rôzne experimenty, každý 5 krát, aby sa vylúčil vplyv počiatočných hodnôt váh a dosiahol sa štatický vierohodný výsledok.

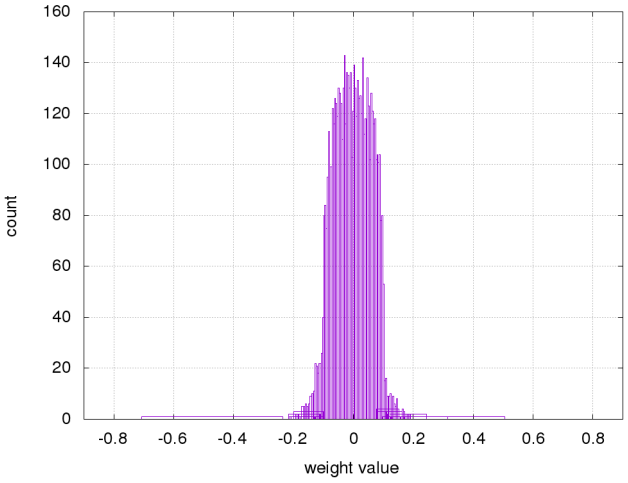
3 Výsledky experimentov

3.1 Riedkosť váh

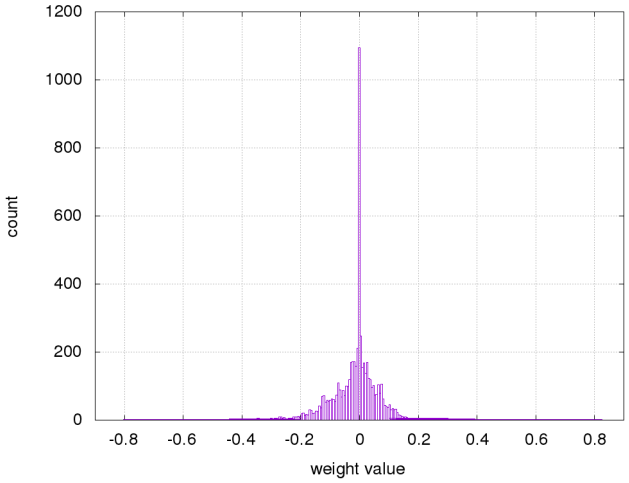
Počas tréningu sa postupne váhy ustália na potrebné hodnoty. Pomocou histogramov je možné overiť riedkosť váh, v závislosti od parametra λ. Pre prehľadnosť sú znázornené len vlastnosti váh pre prvú vrstvu siete. Histogramy na obrázkoch 4 a 5 porovnávajú početnosti váh pre doprednú sieť. Pre riedku sieť je 16.1% váh nulových, pre neriedku 1.97%. Podobné výsledky sú pre kombináciu autoenkódera a doprednej siete - obrázky 6 a 7 S percentuálnym zastúpením 2.44% pre riedke váhy a 17.8% pre neriedke váhy. Ilustračne ešte uvádzame cahrakter riedkych a neriedkych váh pre každý zo 64 neurónov, usporiadaných do matice 5x19 - tak aby zodpovedali vizuálnej interpretácií prostredia. Obrázky 8, 9.



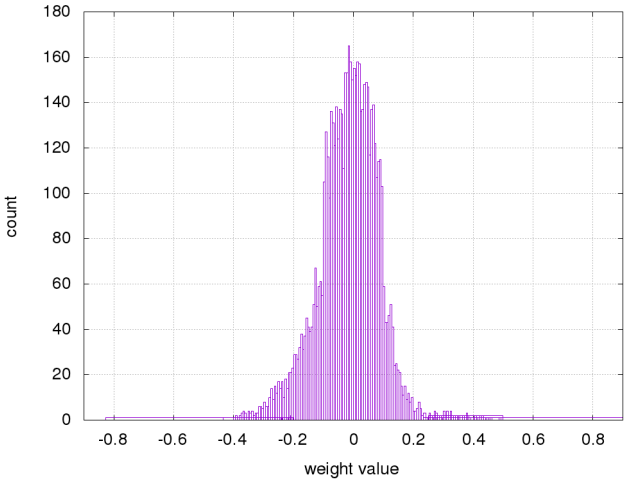
Obr. 4: FNN sparse weights histogram



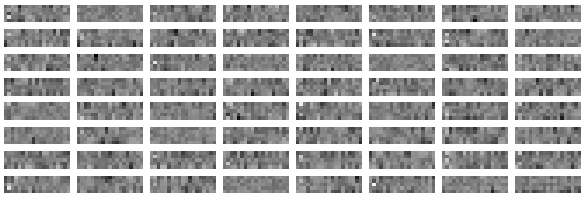
Obr. 5: FNN no sparse weights histogram



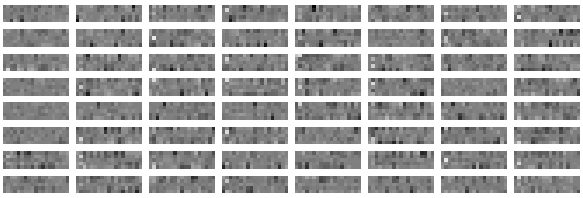
Obr. 6: AE+FNN sparse weights histogram



Obr. 7: AE+FNN no sparse weights histogram



Obr. 8: AE+FNN no sparse weights visualisation



Obr. 9: AE+FNN sparse weights visualisation

3.2 Dosiahnuté skóre

Siet s autoenkóderom sa najpr predtrénovala 100 000 iteráciami. Následne sa spustil supervised tréning. Po 200 000 supervised učiacich iteráciach sa spustilo 50 000 testovacích iterácií. Zhrnuté dosiahnuté skóre na konci testovacích iteráciach je uvedené v tabuľke 2.

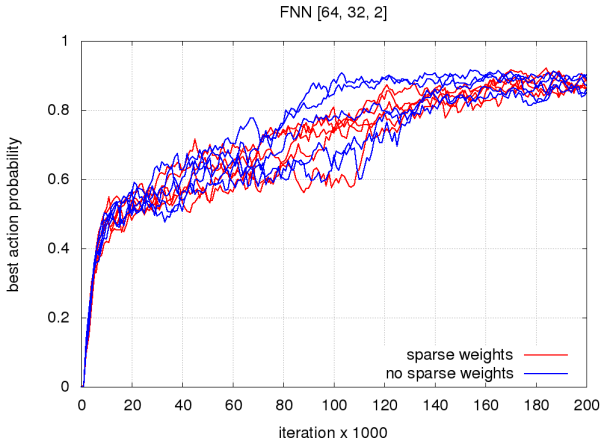
Tabuľka 2: Summary results **TODO prepocitat**

	average score	best score	worst score	average best action probability [%]
FNN sparse weights	957.31	978.3	927.31	94.04
FNN nosparse weights	951.5	959.3	942.644	95.95
AE+FNN sparse weights	763.58	942.97	618.66	88.16
AE+FNN no sparse weights	737.78	884.98	618.99	87.19

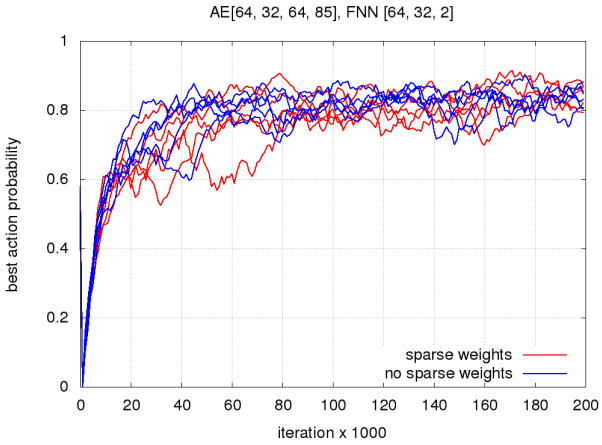
Testovanie hypotézy o rýchlosti učenia zobrazujú grafy na obrázkoch 10, 11, 14. Grafy zobrazujú priebehy pre všetkých 10 sieti (5 riedkych a 5 neriedkych). Graf na obrázku 10 je pre jednoduchú doprednú sieť, a graf na obrázku 11 pre kombinované riešenie s predtrénovaním.

Pre prehľadnosť sme z každého grafu vybrali spriemerovali výsledky sieti a vykreslili ich do spoločného grafu, na obrázku 14. Z grafu obrázka 14 je zrejmé, že predtrénovaná sieť sa učí rýchlejšie - voľbu správnej akcie dosiahne rýchlejšie. Po čase podľa očakávania, obe siete dosiahnú rovnaký výsledok. Hypotéza sa teda potvrdila.

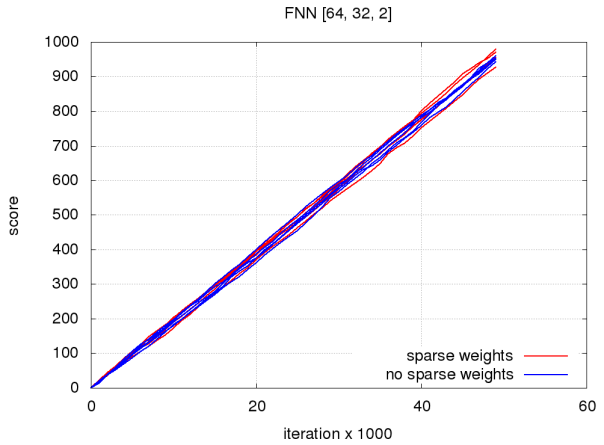
Rovnako nie je viditeľný kvalitatívny rozdiel medzi riedkymi a neriedkými váhami - optimalizáciou harvéru na riedke výpočty by tak mohlo byť výrazne urýchlene. Pre úplnosť dodávame dosiahnuté skóre počas testovania, obrázky 12 a 13. Po dostatočnom počte trénovacích iterácií sa teda všetky uvedené riešenia správajú veľmi podobne.



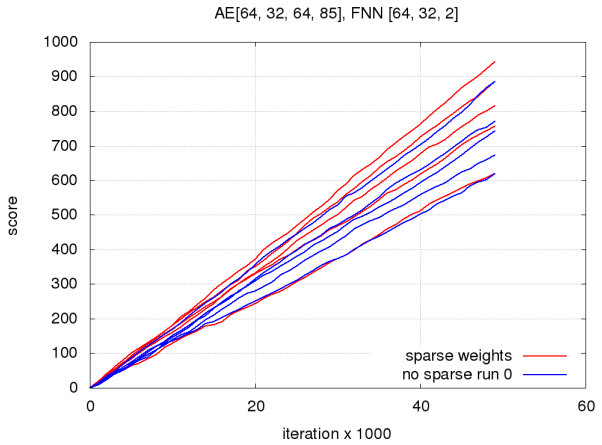
Obr. 10: FNN progress comparison



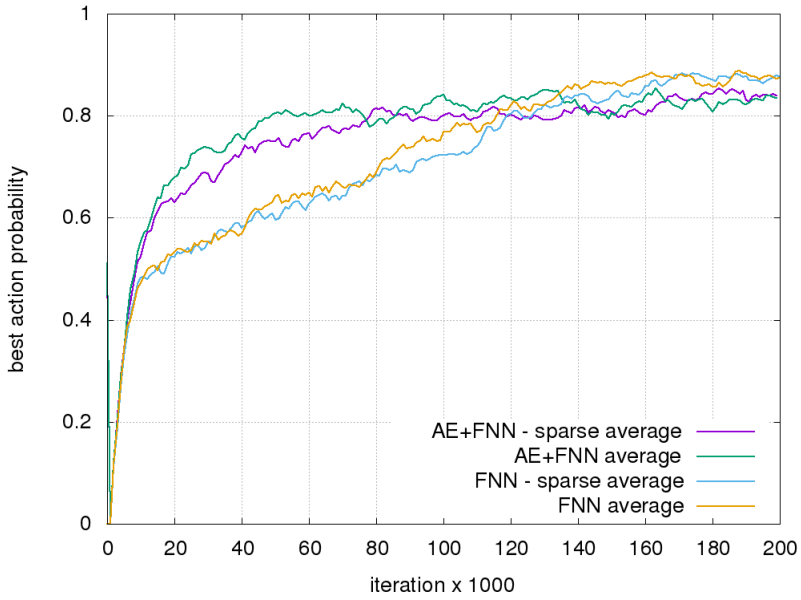
Obr. 11: AE+FNN progress comparison



Obr. 12: FNN score



Obr. 13: AE+FNN score



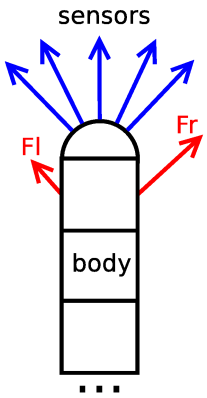
Obr. 14: Average training progress comparison

4 Aditional experiment - snake

Pozitívne výsledky uvedeného experimentu nás viedli urobiť ešte jeden, doplnkový experiment. Použili sme klasickú hru had - agent má za úlohu zbierať jedlo, obrázok 15. Po zjedení jedla sa jedlo objaví na novom mieste. Za každé zjedenie je odmena $+0.1$, za vypadnutie z mapy -1.0. Agent - had, má na hlave 32 senzorov, ktoré ho informujú o vzdialenosti k jedlu. Senzory sú rozmiestnené v polkruhu, tak aby rovnomerne pokrývali zorné pole. Agent má na výber tri akcie - vpred, vľavo, vpravo. Had je modelovaný ako diferenciálny podvozok **TODO citovat** so zotrvačnosťou. Uvedené akcie teda generujú dve sily, ktoré menia polomer zatáčania, tak ako je uvedené na obrázku 16. Kolízie neboli v experimente použité.

Hyperparametre siete boli rovnaké ako v predošlom experimente. Parametre agenta bolo nutné upraviť, pravdepodobnosť voľby horšej akcie $\epsilon = 0.3$, veľkosť batch 100 krokov, a počet epoch batchu 10. S týmito parametrami sa bol agent schopný učiť a za pomerne krátky čas dosahoval kladné odmeny. Kedže úloha ma menej prvkový stavový vektor, zmenšili sme počty neurónov v sietiach. Topológia autoenkódera bola 32, 8, 32, 32 a topológia doprednej siete 32 8 3 [2].

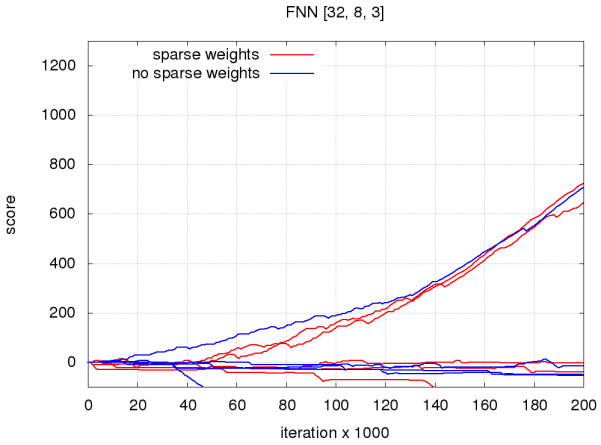
Obr. 15: Testing snake game



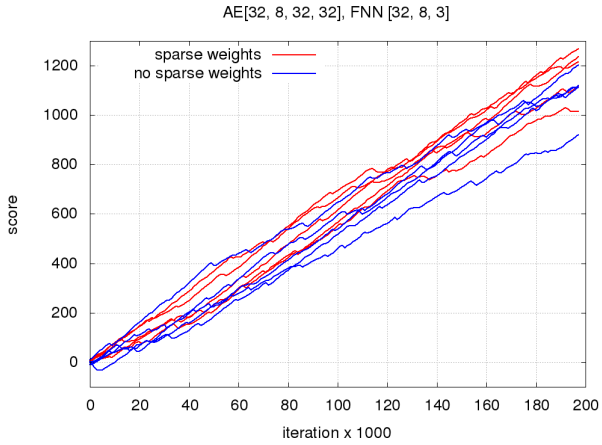
Obr. 16: Agent schematics **OBRAZOK je strasny**

Skóre dosiahnuté počas tréningu je na obrázkoch 17, 18 a 19. Vzhľadom na variabilitu výskytu odmien, sú ropztyly oveľa väčšie ako v predošlom experimente. Prekvapujúce je zistenie, že niektoré dopredné siete nedosiahli kladné skóre ani po 200 000 trénovacií iteráciach. Naproti tomu, všetky predtrénované siete dosiahli kladné skóre, tri z nich prekonali jednoduché dopredné siete. Na obrázku 19 sú znázornené najlepšie siete z oboch topológií pre porovnanie. Naša hypotéza sa teda aj tu potvrdila - pretrénovanie siete naozaj zvýšilo rýchlosť tréningu.

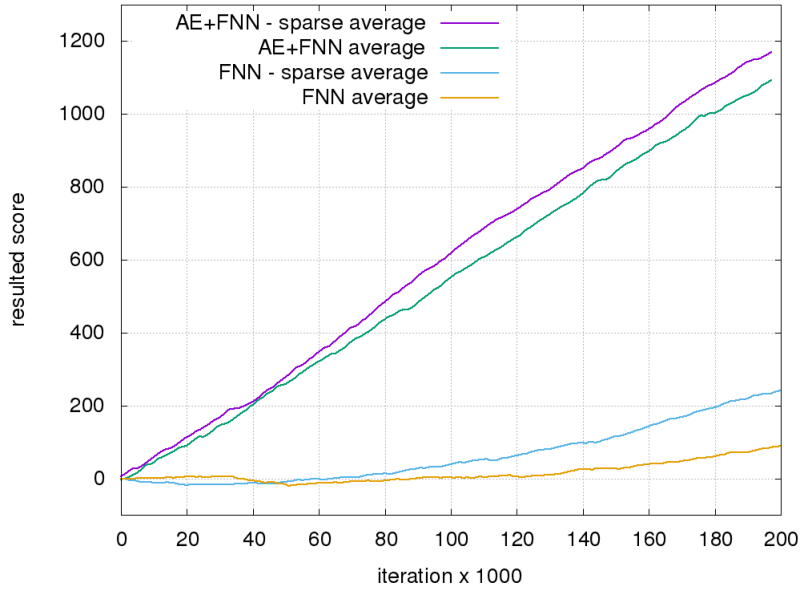
[2] pozri obrázky 2 a 3



Obr. 17: FNN score progress comparison



Obr. 18: AE+FNN score progress comparison



Obr. 19: Training worms score progress for best networks

5 Conclusion

Simulačný nástroj bol napísaný v C++, neurónové siete sú počítané pomocou NVIDIA Cuda technológie. Je možné voliť medzi počítaním na CPU a GPU - výsledné zlepšenie závisí od veľkosti siete. Beh experimentov bol asi 120 minút.

Literatúra

[1] M.J.Flynn, P.Kanerva, and N.Bhadkamkar, 1989, Sparse Distributed Memory: Principles and Operation http://i.stanford.edu/pub/cstr/reports/csl/tr/89/400/CSL-TR-89-400.pdf

[2] David Rogers, 1988, NASA, KANERVA'S SPARSE DISTRIBUTED MEMORY: AN ASSOCIATIVE MEMORY ALGORITHM WELL SUITED TO THE CONNECTION MACHINE https://pdfs.semanticscholar.org/9288/bb551f000348f800ff40d0fdb3fd74c410ef.pdf

[3] J. S. Albus, 1975, Data Storage in Cerebellar Model Articulation Controller https://www.cs.cmu.edu/afs/cs/academic/class/15883-f13/readings/albus-1975.pdf

[4] Olshausen and Field (1997): Sparse coding with an overcomplete basis set

[5] Olshausen BA, Field DJ (2004) : Sparse coding of sensory inputs. Current Opinion in Neurobiology, 14, 481-487

[6] Mushroom body, locust (Laurent)

[7] HVC, zebra finch (Fee)

[8] Auditory cortex, mouse (DeWeese & Zador)

[9] Hippocampus, ratprimate(Thompson & Best; Skaggs)

[10] Motor cortex, rabbit (Swadlow)

[11] Visual cortex, monkeycat (Vinje & Gallant)

[12] Inferotemporal cortex, human (Fried & Koch)