Deep Reinforcement Learning agents playing DOOM

Kashtanova Victoriya and Hurault Samuel

Object Recognition Final Project

January, 15 2019

Visual Doom AI Competition

	Limited Deathmatch		Full Deathmatch	
Agent Name	Number of frags	K/D Ratio	Number of frags	K/D Ratio
5vision	142	0.41	12	0.20
AbyssII	118	0.40	-	-
Arnold	413	**2.45**	164	**33.40**
CLYDE	393	0.94	-	-
ColbyMules	131	0.43	18	0.20
F1	**559**	1.45	-	-
IntelAct	-	-	**256**	3.58
Ivomi	-578	0.18	-2	0.09
TUHO	312	0.91	51	0.95
WallDestroyerXxx	-130	0.04	-9	0.01

Figure: Results of the Visual Doom AI Competition 2016. Scores marked with '-' indicate that the agent did not participate in the corresponding track. The best results in each column are marked in bold[1].

[1] Devendra Singh Chaplot and Guillaume Lample. "Arnold: An Autonomous Agent to Play FPS Games". In: *AAAI*. 2017.

Project objectives

- 2 methods :
  - Learning To Act by Prediction the Future (**DFP**)[2]
  - Playing FPS Games with Deep Reinforcement Learning (**Arnold**)[3]
- Replicates each article's main results in Doom
- Optimize the methods
- Evaluation of the methods in an other environment

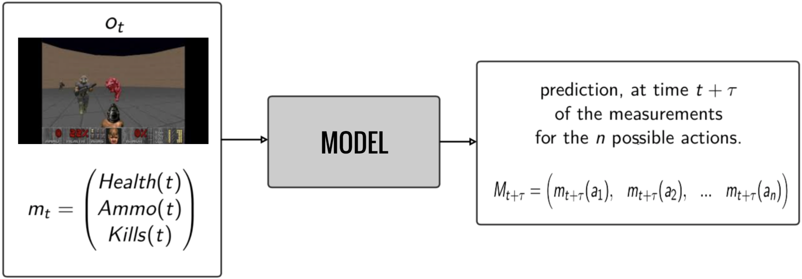
[2] Alexey Dosovitskiy and Vladlen Koltun. "Learning to Act by Predicting the Future". In: *CoRR* abs/1611.01779 (2016). arXiv: 1611.01779. URL: http://arxiv.org/abs/1611.01779.

[3] Guillaume Lample and Devendra Singh Chaplot. "Playing FPS Games with Deep Reinforcement Learning.". In: *Proceedings of AAAI*. 2017.

Introduction to the DFP model

Learning To Act by Prediction the Future

At each game time step t : predict future measurements



Introduction to the DFP model

o_t

$$m_t = \begin{pmatrix} Health(t) \\ Ammo(t) \\ Kills(t) \end{pmatrix}$$

MODEL

prediction, at time $t+\tau$
of the measurements
for the n possible actions.

$$M_{t+\tau} = \begin{pmatrix} m_{t+\tau}(a_1), & m_{t+\tau}(a_2), & \dots & m_{t+\tau}(a_n) \end{pmatrix}$$

goal vector $g = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix}$

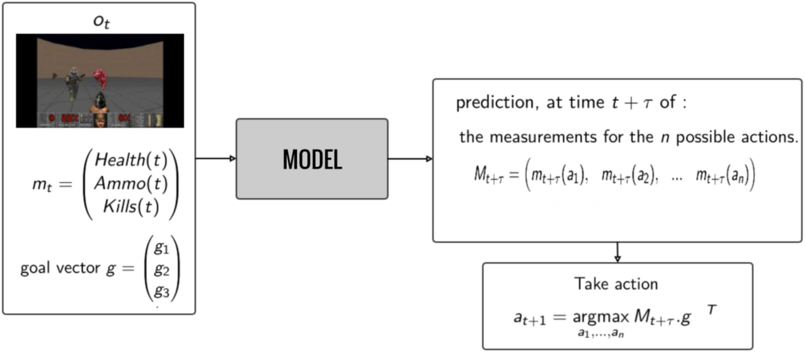
Take action

$$a_{t+1} = \underset{a_1,\dots,a_n}{\operatorname{argmax}} M_{t+\tau}.g^T$$

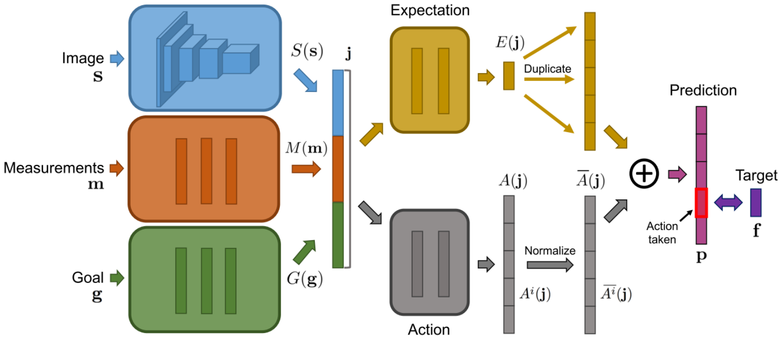
Introduction to the DFP model

We want to specify which measurements we care about at any given time

At each game time step t :



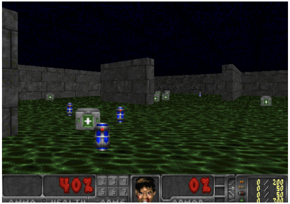
The model



- No scalar reward.
- Trained on experiences previously collected : **Supervised learning**
- Predict future measurement for 3 different future time steps $\tau = (8, 16, 32)$.

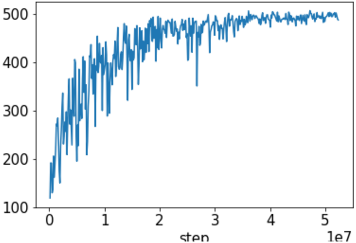
Experiments

Two given scenarios :

Name	Health gathering	Battle
Image		
Nb Actions	4	8
Measurements	(Health)	(Ammo,Health,Kills)

Health Gathering scenario

- Basic training from the article : episode limited to 525 steps.



- Training with longer episodes : episode limited to 2100 steps.

Testing \ Training	short episodes	long episodes
long episodes	658	**1166**

Figure: Life time (Number of step of an episode)

Health Gathering scenario

Battle scenario

- Goal vector input random in $[-1, 1]$ during learning.

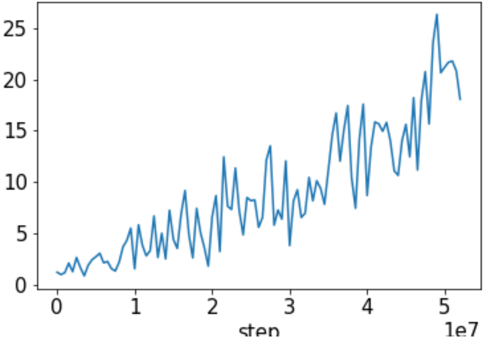


Figure: Kill/Death ratio during learning

Battle scenario

- Training with short and long episodes

Testing \ Training	short episodes	long episodes
unlimited episodes	2263	**3690**

Figure: Life time (Number of step of an episode)

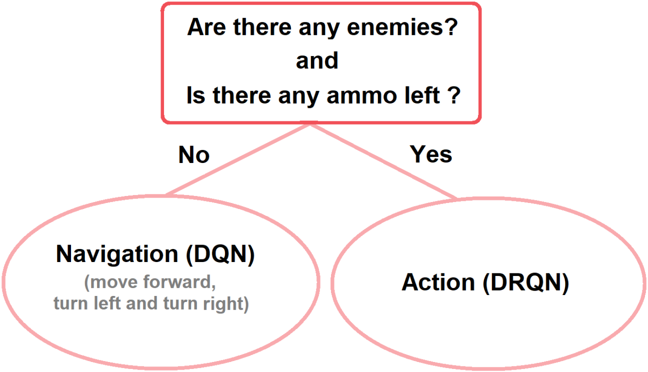
- Choice of the input goal vector at inference time ($Ammo$, $Health$, $Kills$).

Testing \ Training	Random goal in $[-1, 1]$
(0.5, 0.5, 1)	**35.1**
$(1, 1, 1)$	27.2
$(0, 0, 1)$	3.2

Figure: Kill / Death ratio

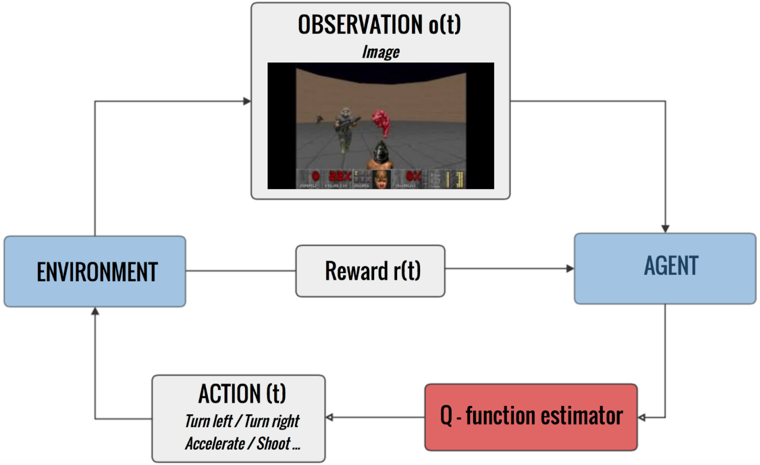
Battle scenario

Arnold's model[4]

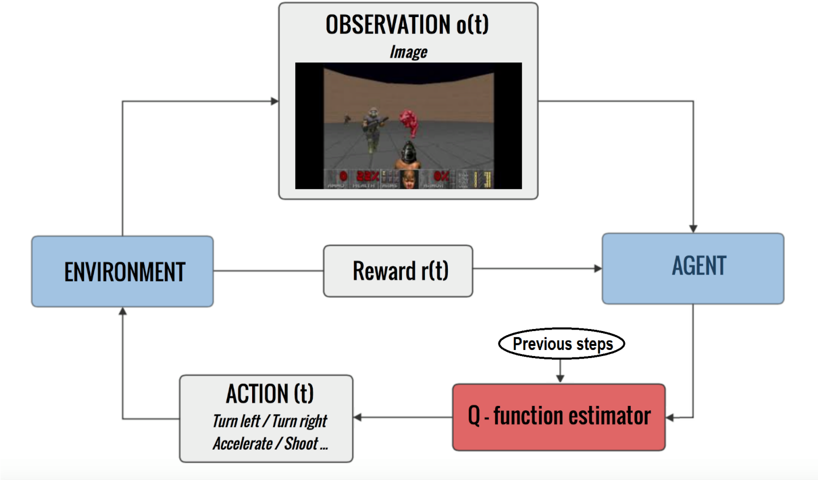


[4] Guillaume Lample and Devendra Singh Chaplot. "Playing FPS Games with Deep Reinforcement Learning.". In: *Proceedings of AAAI*. 2017.

Deep Q-Networks (Navigation)



Deep Recurrent Q-Networks (Action)



Deep Recurrent Q-Networks (Action)

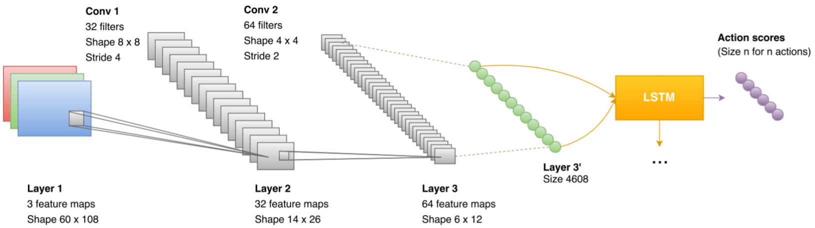


Figure: Initial DRQN model[5].

[5] Matthew J. Hausknecht and Peter Stone. "Deep Recurrent Q-Learning for Partially Observable MDPs". In: *AAAI Fall Symposia*. 2015.

Deep Recurrent Q-Networks (Action)

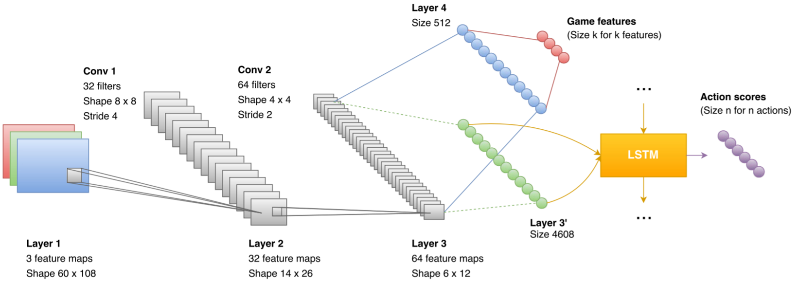


Figure: DRQN model with features[6].

[6] Guillaume Lample and Devendra Singh Chaplot. "Playing FPS Games with Deep Reinforcement Learning.". In: *Proceedings of AAAI*. 2017.

Experiments : Deathmatch

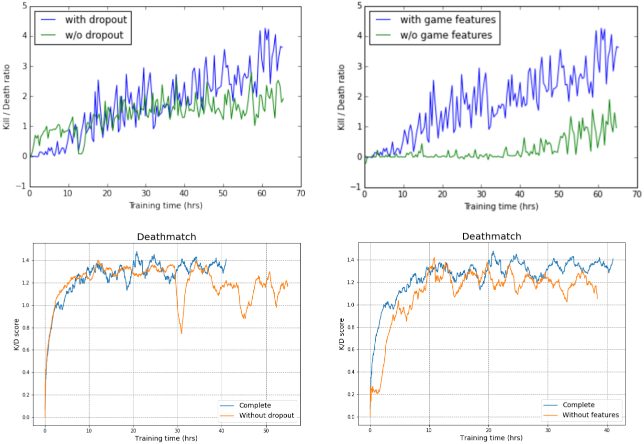
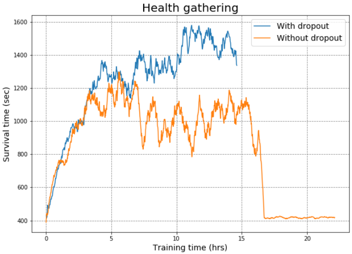


Figure: Plot of K/D score of action network on limited deathmatch as a function of training

Deathmatch Video

Experiments : Health gathering



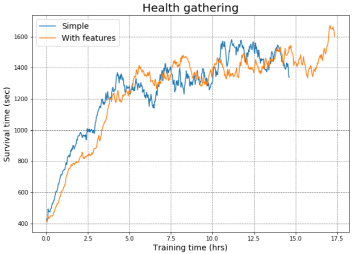
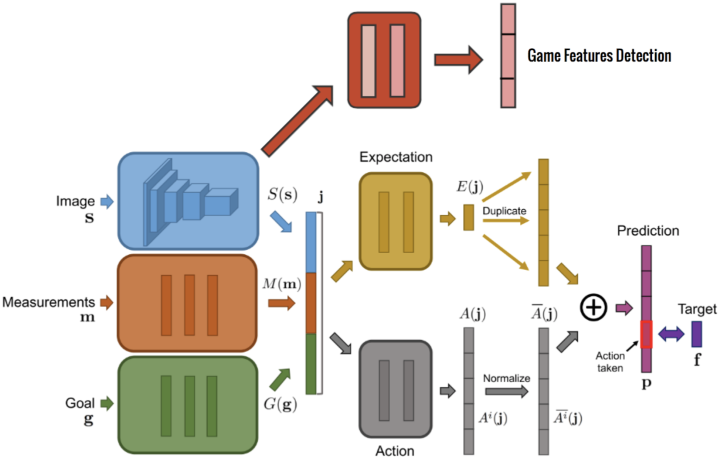


Figure: Plot of average Survival time on Health gathering supreme as a function of training

Use game features information on DFP



Experiment

Scenario : Health gathering Game features $= \{medikit, poison\}$

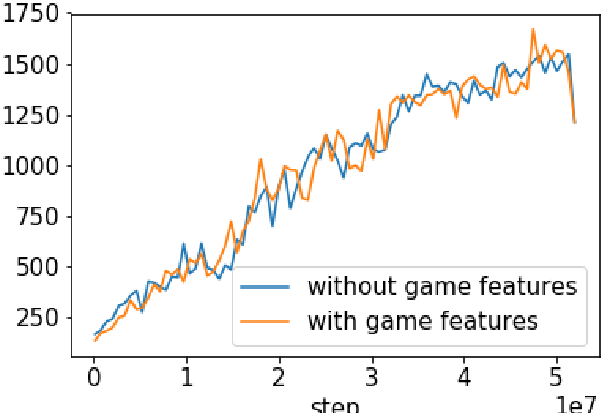


Figure: Life time during training with and without item detection

Experiment

Scenario : Battle // Learning with random goal in $[-1, 1]$, testing with fixed goal (0.5,0.5,1). Game features = $\{enemy\}$

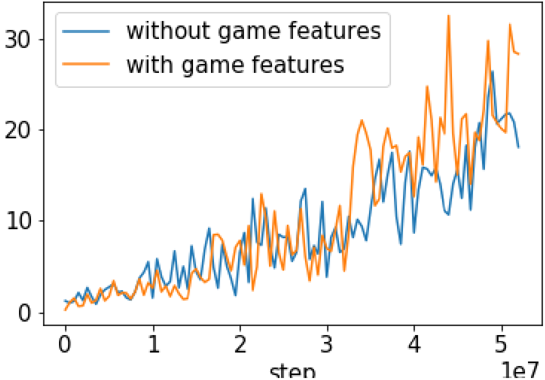


Figure: Kill / Death ratio during training with and without enemy detection.

Comparison : Health gathering

Both methods learned on the very same scenario.

	DFP	Arnold
Life time (nb of steps)	4664	1283

Comparison : Defend the center

Methods learned on different battle scenarios.

	DFP	Arnold
Kill/Death	8.9	8.6

What we have done ...

- Comparison of two different RL formulations : Q-learning (Arnold) vs Supervised Learning (DFP).
- Replicated the main results of both articles.
- Improved the DFP network with ideas from the Q-learning network.

To go further ...

- Optimize the parameters.
- Use Arnold navigation / action network split on the DFP method.
- Make them play against each other
- Adapt to an other 3D environment : CARLA (autonomous driving) and MINOS (Indoor navigation).