Deep Reinforcement Learning

Today’s AIs that beat humans

Paulo Bruno Serafim

DataPeste - CorongaMeet 2.0
2020

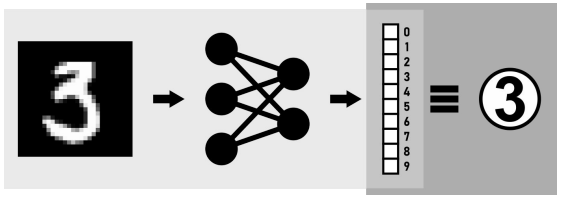
paulobruno.github.io

CRIA

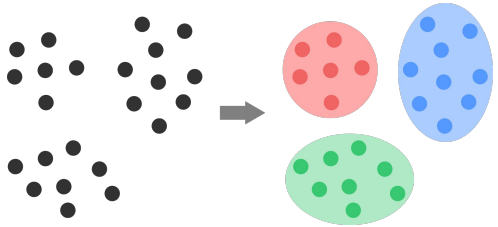
MDCC UFC

Atlântico

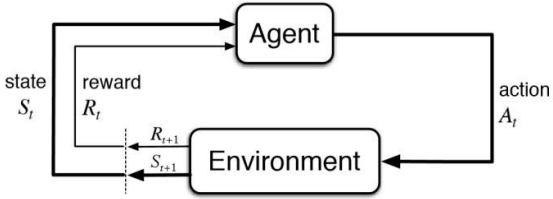
Reinforcement Learning



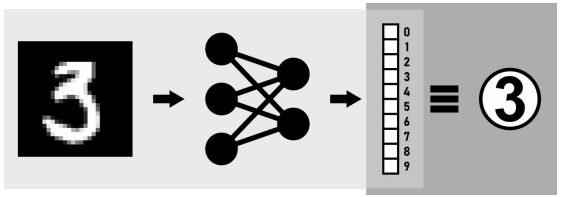
Supervised Learning



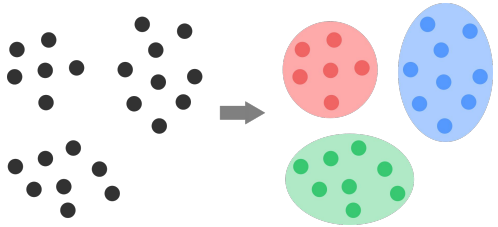
Unupervised Learning



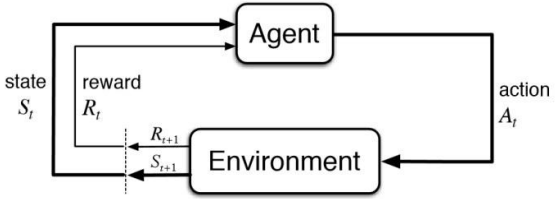
Reinforcement Learning



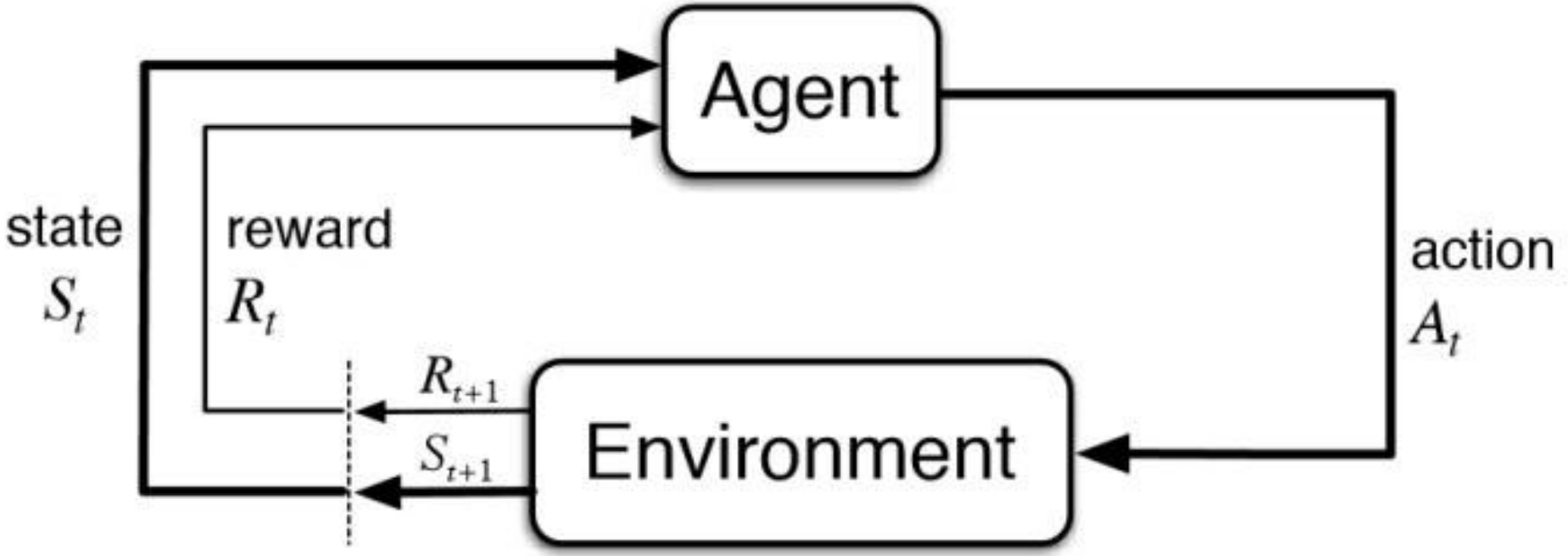
Supervised Learning



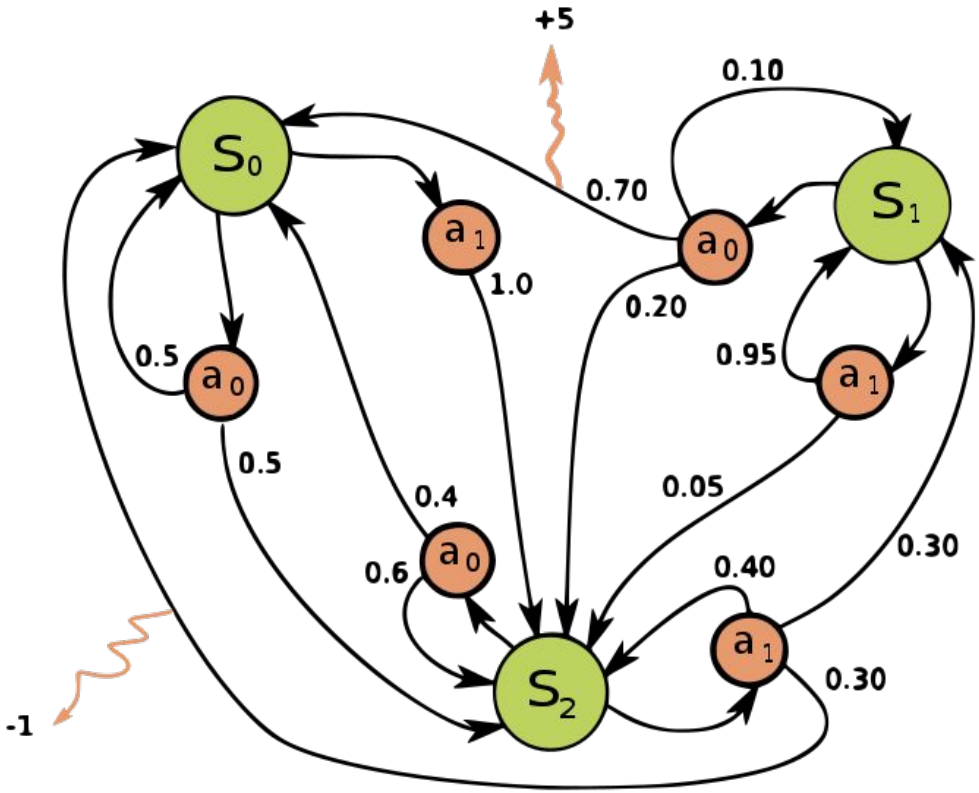
Unupervised Learning



Reinforcement Learning

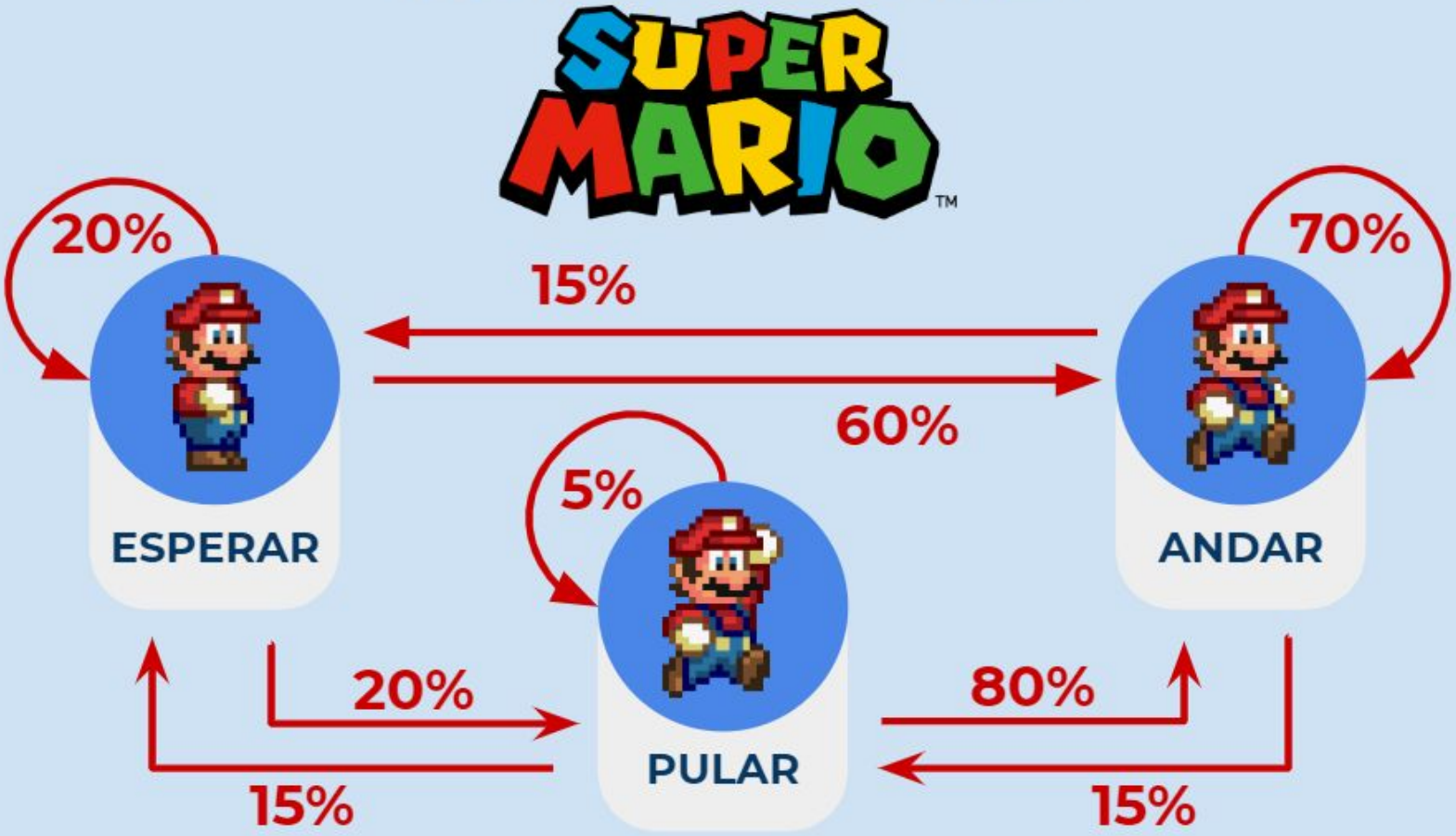


Sutton and Barto. Reinforcement Learning. 2018.

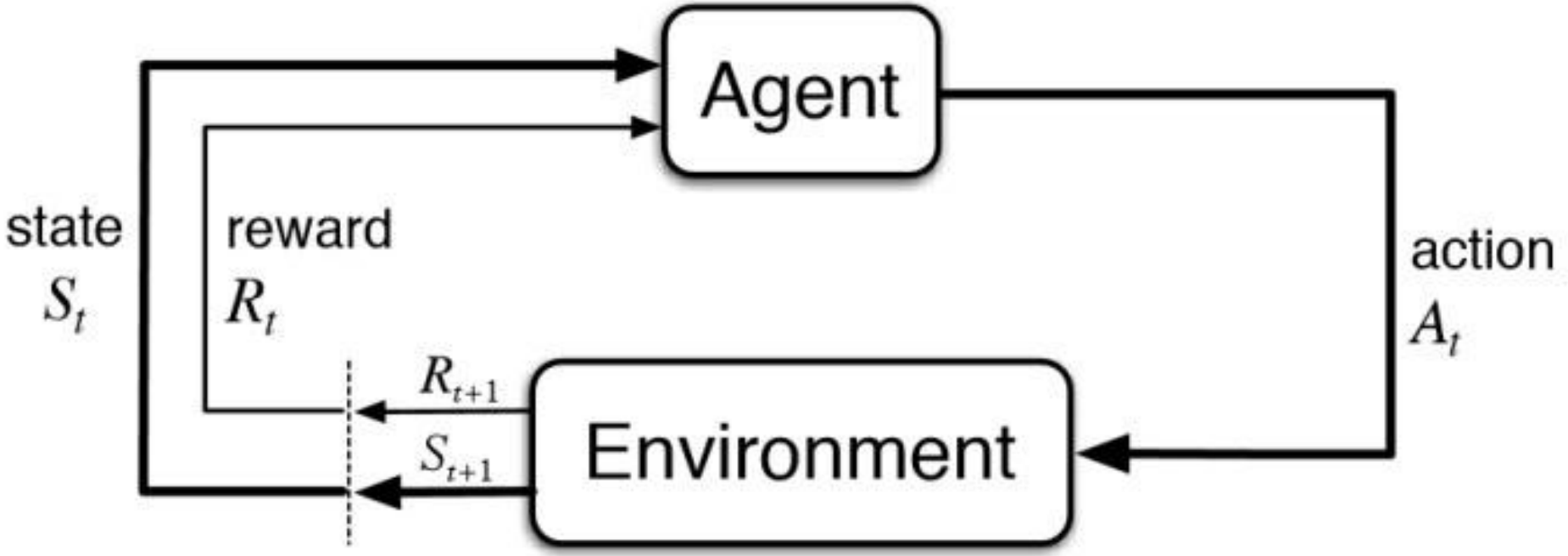


https://en.wikipedia.org/wiki/Markov_decision_process

PROCESSO DE MARKOV



https://paulovasconcellos.com.br/explicando-deep-reinforcement-learning-com-super-mario-ao-inv%C3%A9s-de-matem%C3%A1tica-4c77392cc733



Sutton and Barto. Reinforcement Learning. 2018.

Q-Learning

Before learning begins, Q is initialized to a possibly arbitrary fixed value (chosen by the programmer). Then, at each time t th
reward r_t, enters a new state s_{t+1} (that may depend on both the previous state s_t and the selected action), and Q is update
value iteration update, using the weighted average of the old value and the new information:

$$Q^{new}(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$

where r_t is the reward received when moving from the state s_t to the state s_{t+1}, and α is the learning rate ($0 < \alpha \leq 1$).

An episode of the algorithm ends when state s_{t+1} is a final or *terminal state*. However, Q-learning can also learn in non-epis
factor is lower than 1, the action values are finite even if the problem can contain infinite loops.

Wikipedia

Q-Learning

Action value

Learning rate

Reward

Discount factor

Old value

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t))$$

New value

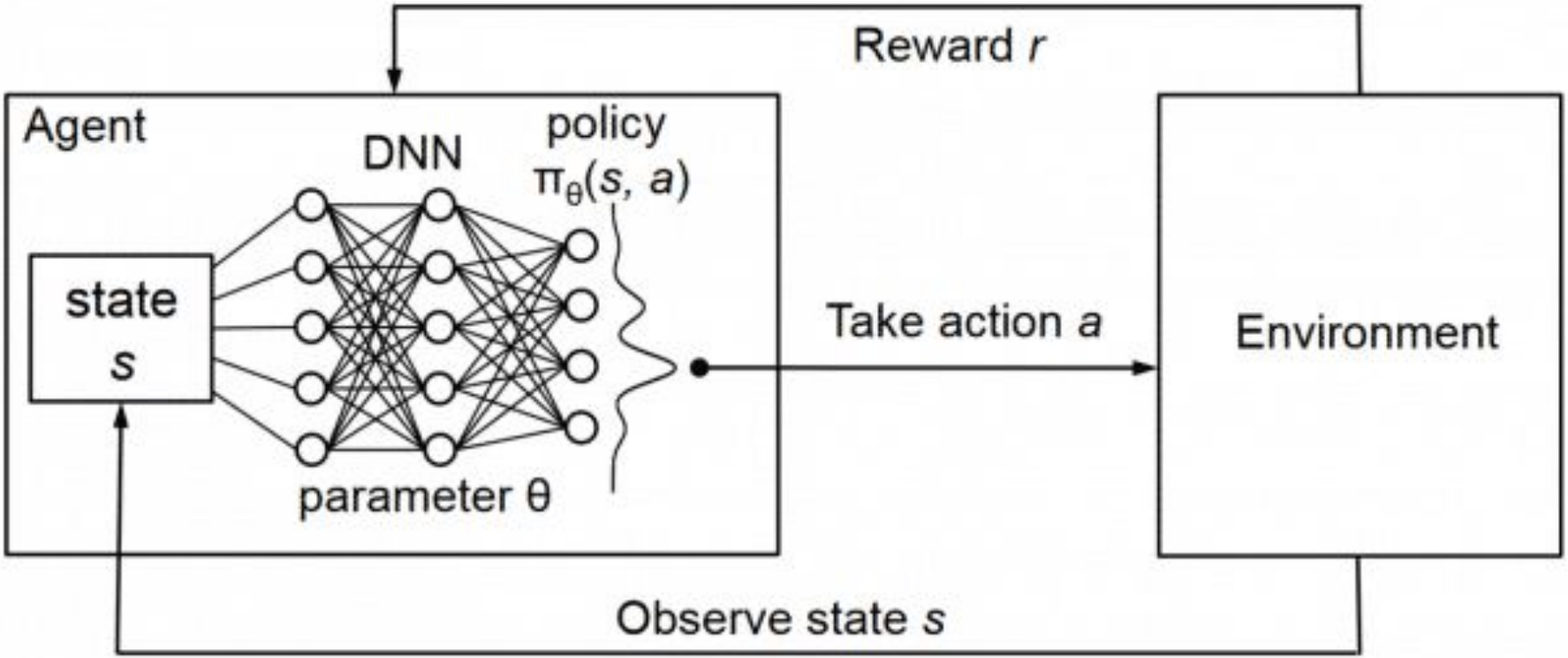
Old value

Estimate of optimal future action value

Code time!

https://colab.research.google.com/drive/173XIDXa1q85DbD1TDlNaNi1-5GqmjMku?usp=sharing

Deep Reinforcement Learning



http://people.csail.mit.edu/hongzi/content/publications/DeepRM-HotNets16.pdf

Playing Atari with Deep Reinforcement Learning

Volodymyr Mnih **Koray Kavukcuoglu** **David Silver** **Alex Graves** **Ioannis Antonoglou**

Daan Wierstra **Martin Riedmiller**

DeepMind Technologies

{vlad,koray,david,alex.graves,ioannis,daan,martin.riedmiller} @ deepmind.com

Abstract

We present the first deep learning model to successfully learn control policies directly from high-dimensional sensory input using reinforcement learning. The model is a convolutional neural network, trained with a variant of Q-learning, whose input is raw pixels and whose output is a value function estimating future rewards. We apply our method to seven Atari 2600 games from the Arcade Learning Environment, with no adjustment of the architecture or learning algorithm. We find that it outperforms all previous approaches on six of the games and surpasses a human expert on three of them.

LETTER

doi:10.1038/nature14236

Human-level control through deep reinforcement learning

Volodymyr Mnih[1]*, Koray Kavukcuoglu[1]*, David Silver[1]*, Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

The theory of reinforcement learning provides a normative account[1], deeply rooted in psychological[2] and neuroscientific[3] perspectives on animal behaviour, of how agents may optimize their control of an environment. To use reinforcement learning successfully in situations approaching real-world complexity, however, agents are confronted with a difficult task: they must derive efficient representations of the environment from high-dimensional sensory inputs, and use these to generalize past experience to new situations. Remarkably, humans and other animals seem to solve this problem through a harmonious combination of reinforcement learning and hierarchical sensory processing systems[4,5], the former evidenced by a wealth of neural data revealing notable parallels between the phasic signals emitted by dopaminergic neurons and temporal difference reinforcement learning algorithms[3]. While reinforcement learning agents have achieved some successes in a variety of domains[6–8], their applicability has previously been limited to domains in which useful features can be handcrafted, or to domains with fully observed, low-dimensional state spaces. Here we use recent advances in training deep neural networks[9–11] to

agent is to select actions in a fashion that maximizes cumulative future reward. More formally, we use a deep convolutional neural network to approximate the optimal action-value function

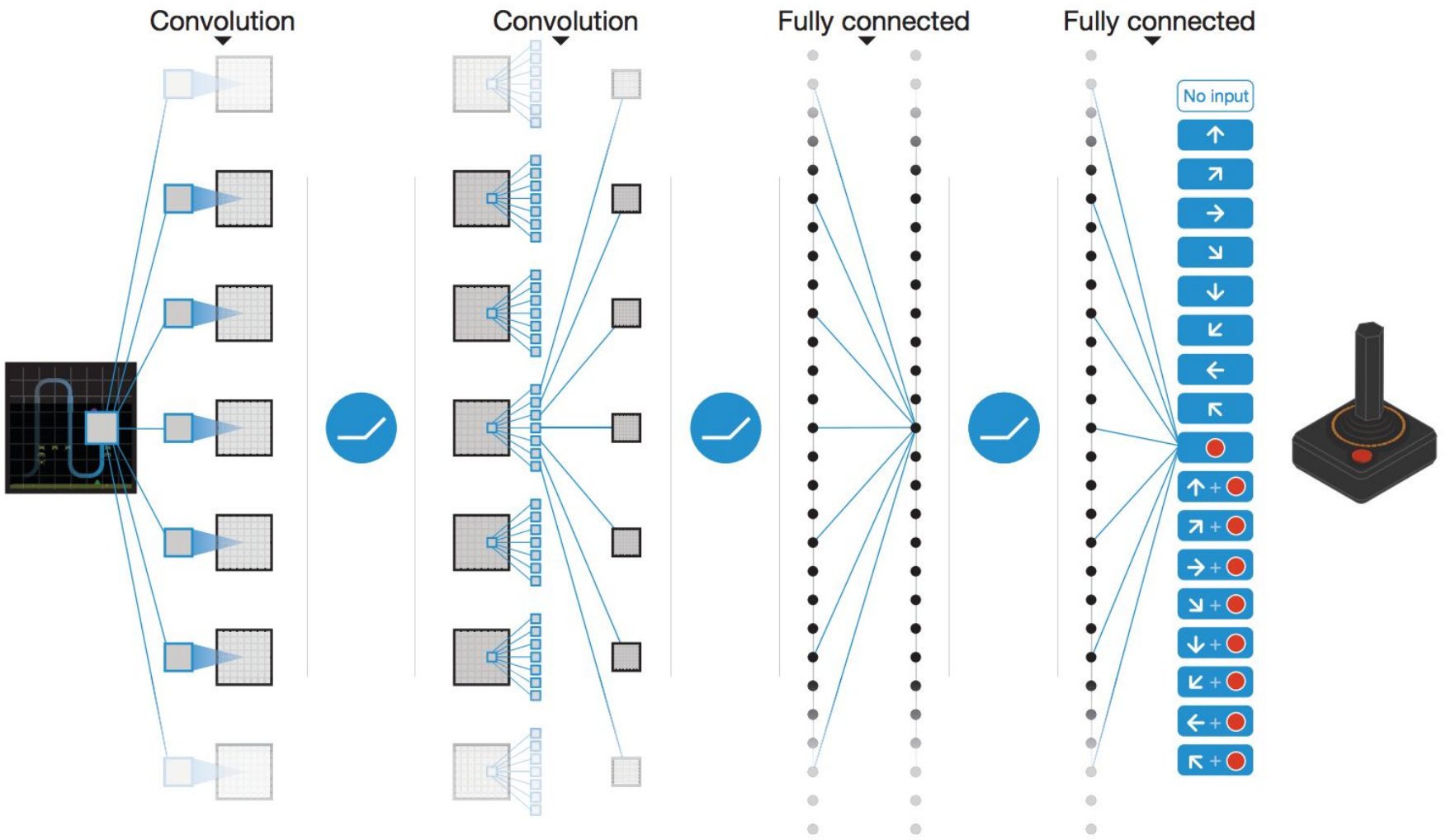
$$Q^*(s,a) = \max_{\pi} \mathbb{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots \mid s_t = s,\, a_t = a,\, \pi\right],$$

which is the maximum sum of rewards r_t discounted by γ at each time-step t, achievable by a behaviour policy $\pi = P(a|s)$, after making an observation (s) and taking an action (a) (see Methods)[19].

Reinforcement learning is known to be unstable or even to diverge when a nonlinear function approximator such as a neural network is used to represent the action-value (also known as Q) function[20]. This instability has several causes: the correlations present in the sequence of observations, the fact that small updates to Q may significantly change the policy and therefore change the data distribution, and the correlations between the action-values (Q) and the target values $r + \gamma \max_{a'} Q(s', a')$. We address these instabilities with a novel variant of Q-learning, which uses two key ideas. First, we used a biologically inspired mechanism termed experience replay[21–23] that randomizes over the data, thereby

https://youtu.be/TmPfTpjtdgg

Mnih et al. (2015)


Convolution
Convolution
Fully connected
Fully connected
No input

AlphaGo *4* vs *1* Lee Sedol

ALPHAGO

TRIBECA FILM FESTIVAL 2017

BFI LONDON FILM FESTIVAL OFFICIAL SELECTION 2017

WINNER TRAVERSE CITY FILM FESTIVAL 2017

OFFICIAL SELECTION MILL VALLEY FILM FESTIVAL 2017

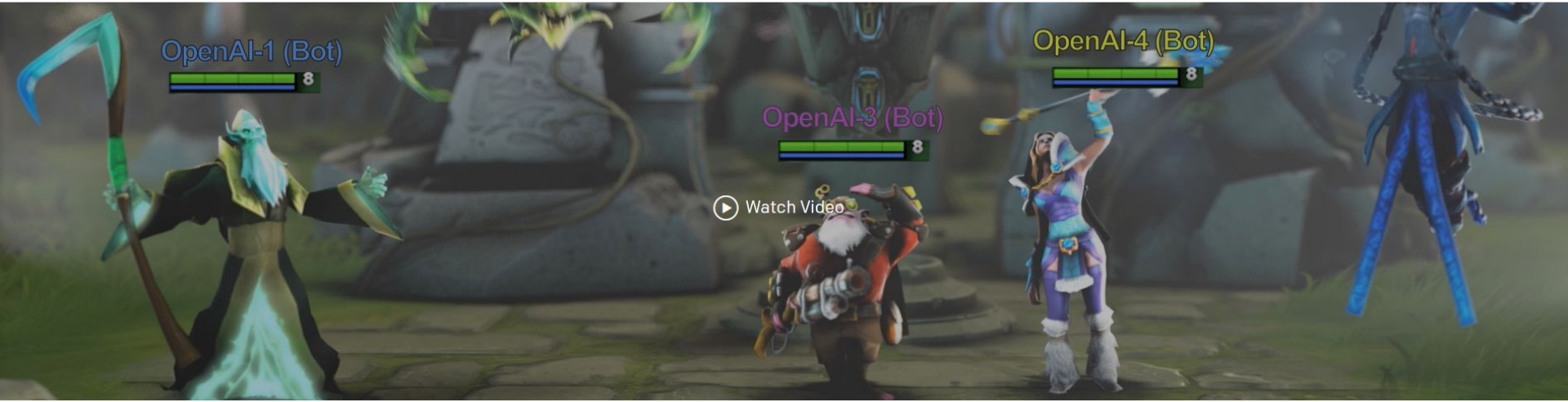
Official Selection WARSAW FILM FESTIVAL

VIFF Official Selection 2017

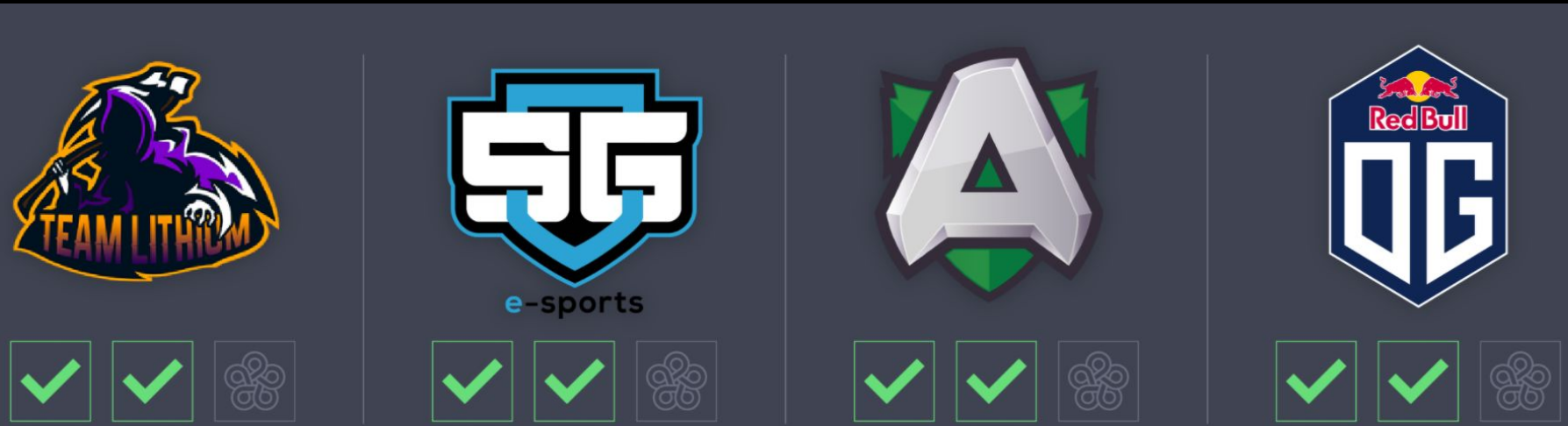
JUNE 25, 2018 • 13 MINUTE READ

OpenAI Five

Our team of five neural networks, OpenAI Five, has started to defeat amateur human teams at Dota 2.



Our team of five neural networks, OpenAI Five, has started to defeat amateur human teams at Dota 2. While today we play with restrictions, we aim to beat a team of top professionals at The International in August subject only to a limited set of heroes. We may not succeed: Dota 2 is one of the most popular and complex



OpenAI Five's record versus semi-pro team Lithium and pro teams SG esports, Alliance, and OG since our losses at The International.

APRIL 15, 2019 • 7 MINUTE READ

OpenAI Five Defeats Dota 2 World Champions

OpenAI Five is the first AI to beat the world champions in an esports game, having won two back-to-back games versus the world champion Dota 2 team, OG, at Finals this weekend. Both OpenAI Five and DeepMind's AlphaStar had previously beaten good pros privately but lost their live pro matches, making this also the first time an AI has beaten esports pros

https://youtu.be/UZHTNBMAfAA

BLOG POST
RESEARCH

24 JAN 2019

AlphaStar: Mastering the Real-Time Strategy Game StarCraft II

DeepMind | StarCraft II

DEMONSTRATION

TLO

Before the match:
"If they can already beat me, that would be incredible." [1]

BLIZZARD ENTERTAINMENT

DeepMind | StarCraft II

DEMONSTRATION

TLO

ROUND	REPLAY
1.	ALPHASTAR WINS
2.	ALPHASTAR WINS
3.	ALPHASTAR WINS
4.	ALPHASTAR WINS
5.	ALPHASTAR WINS

SCORE TLO 0 - 5 ALPHASTAR

BLIZZARD ENTERTAINMENT

DeepMind | StarCraft II

DEMONSTRATION

GRZEGORZ 'MANA' KOMINCZ

Before the match:

"I'm hoping for a 5-0, not to lose any games, but I think the realistic goal would be 4-1 in my favor." [2]

BLIZZARD ENTERTAINMENT

DeepMind | StarCraft II

DEMONSTRATION

GRZEGORZ 'MANA' KOMINCZ

ROUND	REPLAY
1.	ALPHASTAR WINS
2.	ALPHASTAR WINS
3.	ALPHASTAR WINS
4.	ALPHASTAR WINS
5.	ALPHASTAR WINS

SCORE MANA 0 - 5 ALPHASTAR

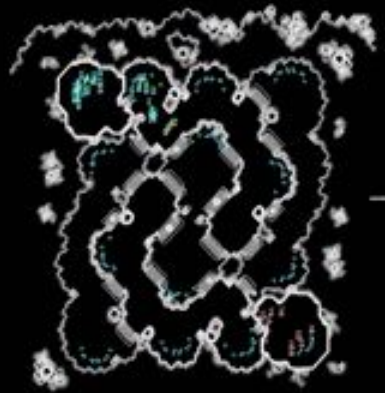
BLIZZARD ENTERTAINMENT

BLOG POST
RESEARCH

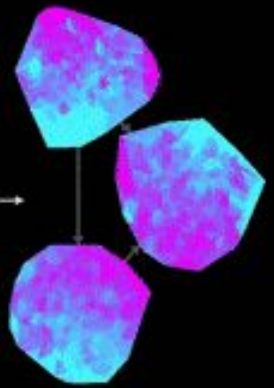
30 OCT 2019

AlphaStar: Grandmaster level in StarCraft II using multi-agent reinforcement learning

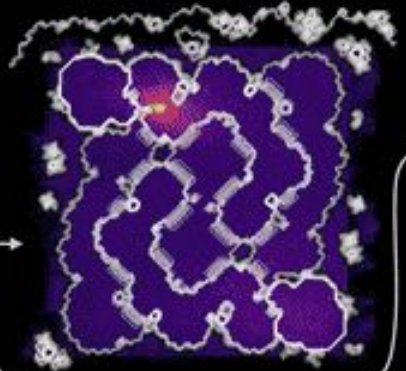
Raw Observations



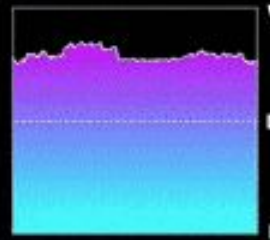
Neural Network Activations



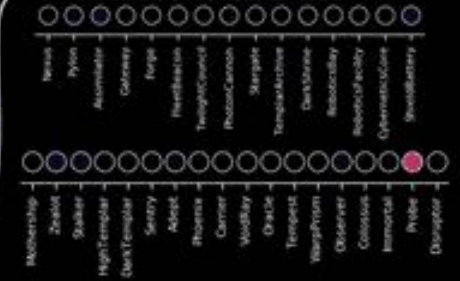
Considered Location

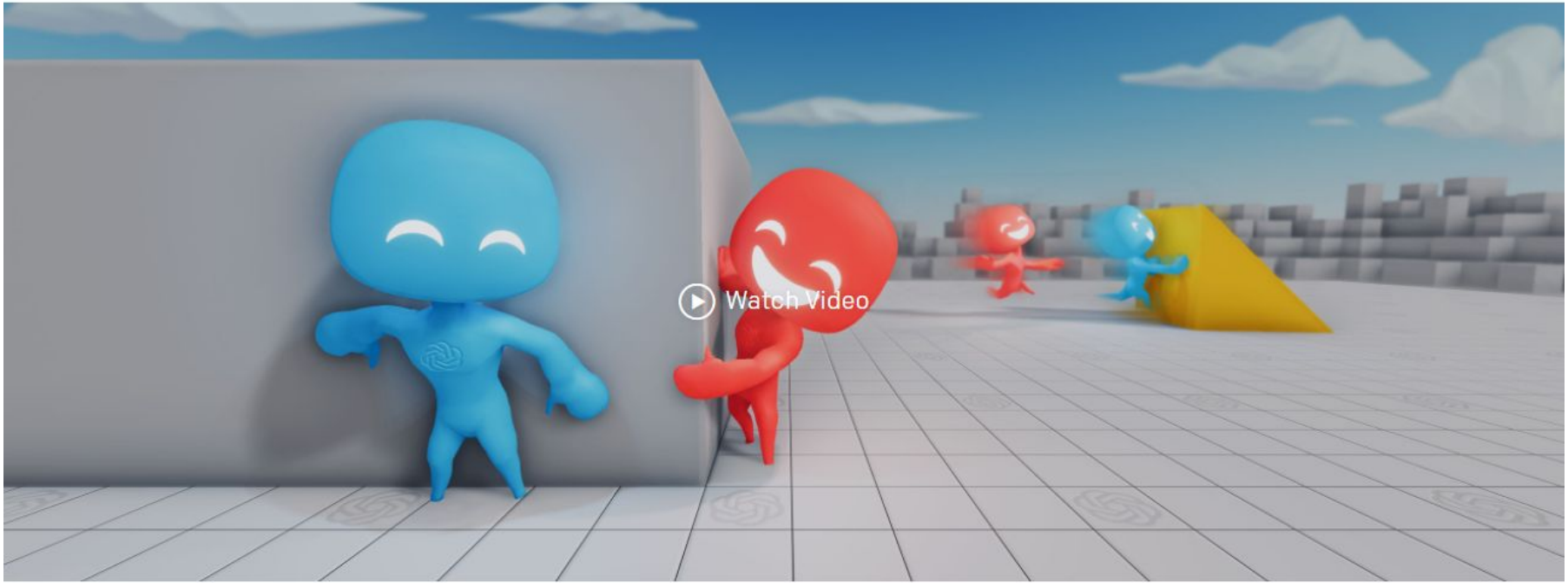


Outcome Prediction



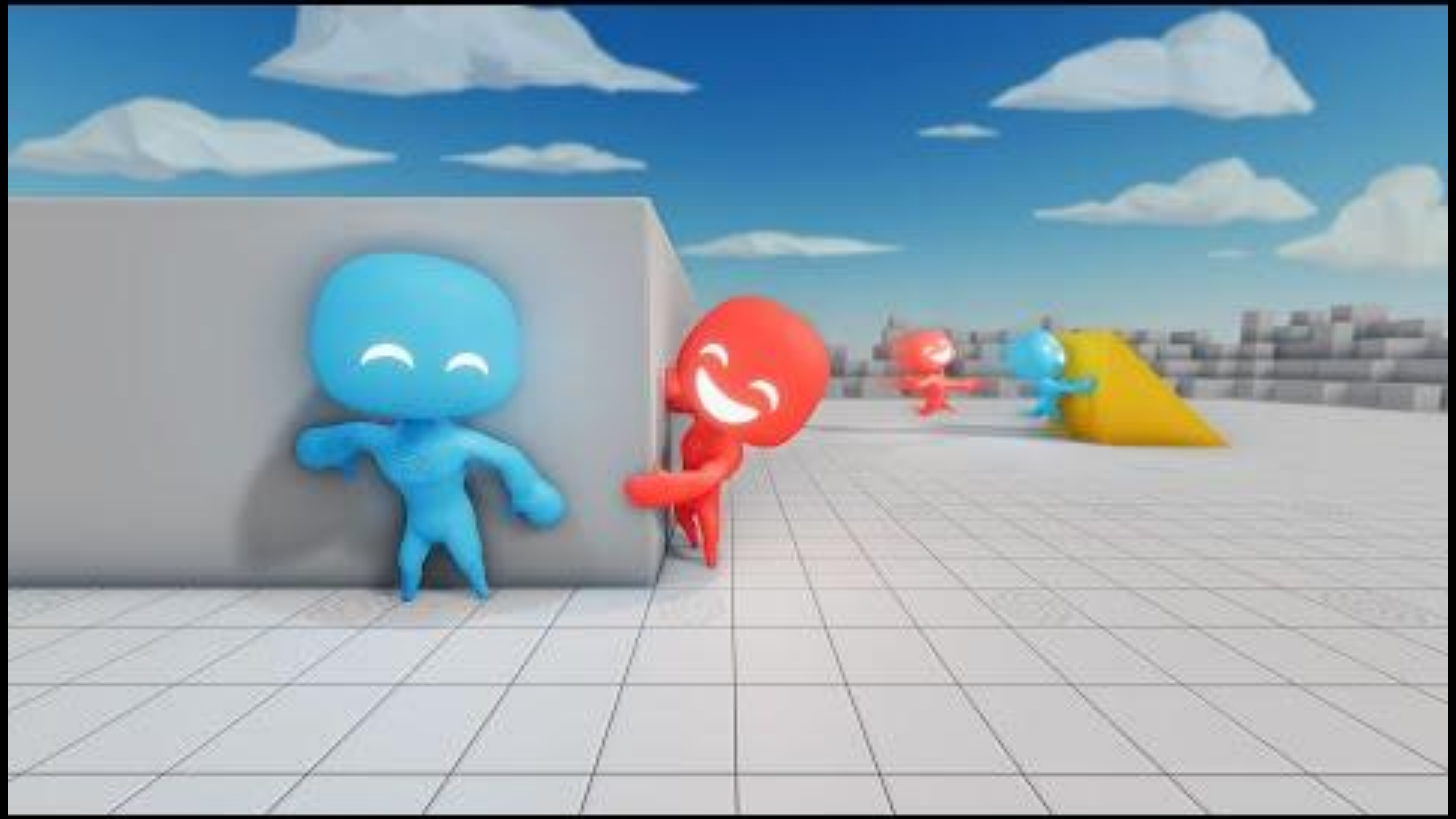
Considered Build/Train

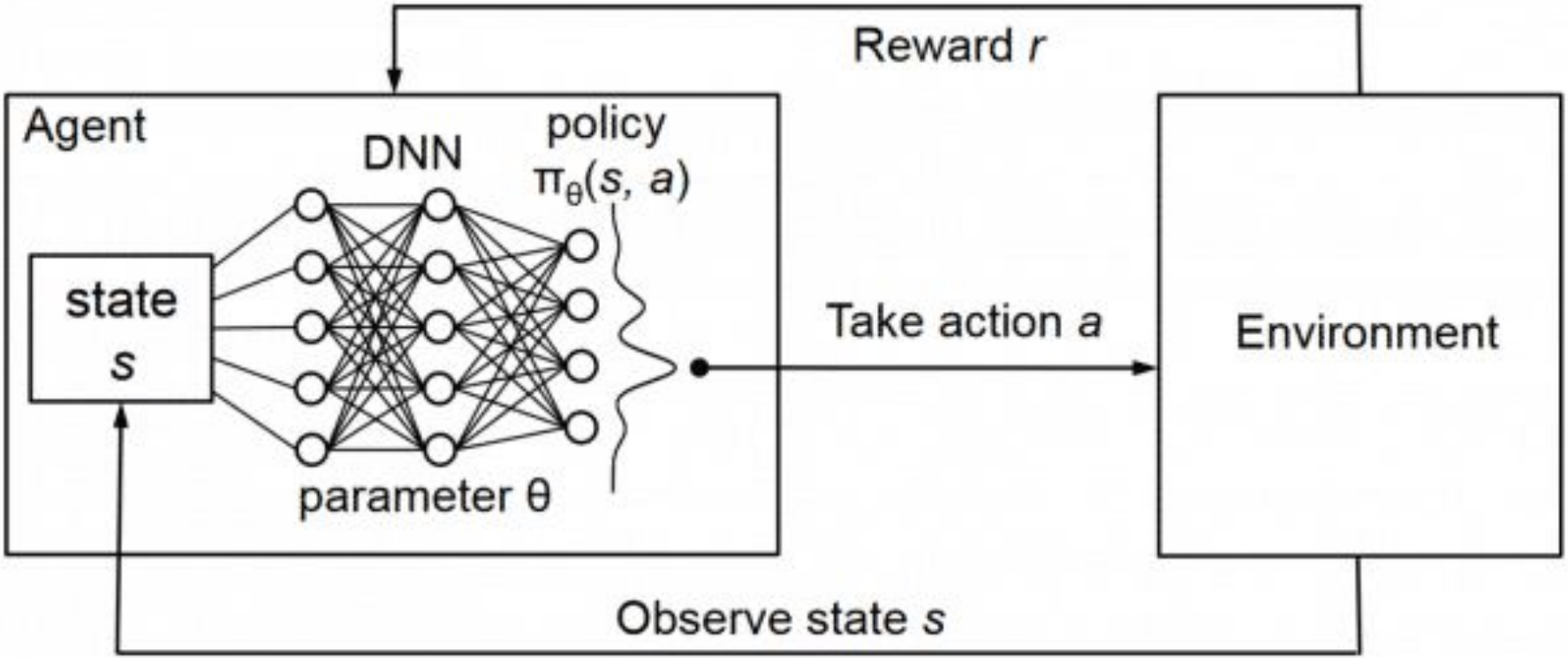




SEPTEMBER 17, 2019 • 9 MINUTE READ

Emergent Tool Use from Multi-Agent Interaction

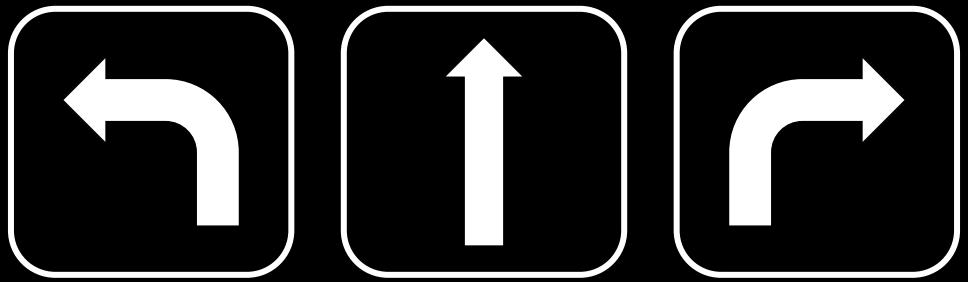


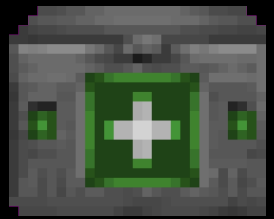


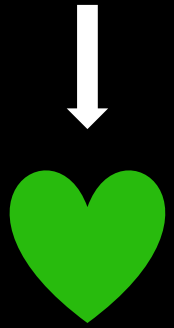
http://people.csail.mit.edu/hongzi/content/publications/DeepRM-HotNets16.pdf


60%
AMMO
HEALTH
2 3 4
5 6 7
ARMS
0%
ARMOR
0 / 200
0 / 50
0 / 50
0 / 300


60%
AMMO
HEALTH
ARMS
0%
ARMOR
0 / 200
0 / 50
0 / 50
0 / 300








60%
AMMO
HEALTH
ARMS
0%
ARMOR
200
50
50
300

https://youtu.be/g6E_c2DgB24

Serafim (2018)

https://youtu.be/LVtxSdJiW4c

Serafim (2018)

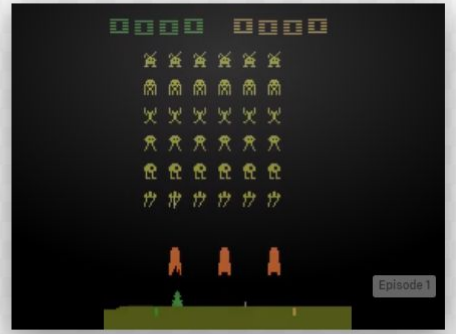
Reinforcement Learning
Environments

Gym

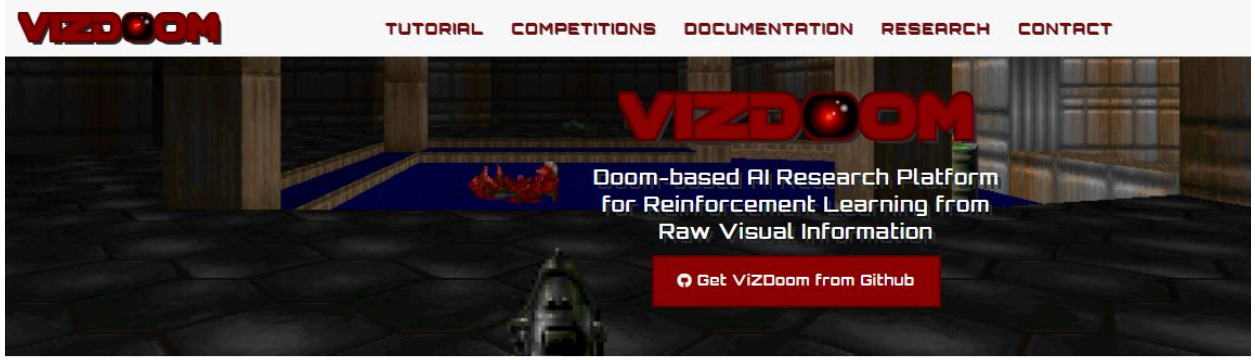
Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Pinball.

View documentation ›
View on GitHub ›

RandomAgent on Pendulum-v0

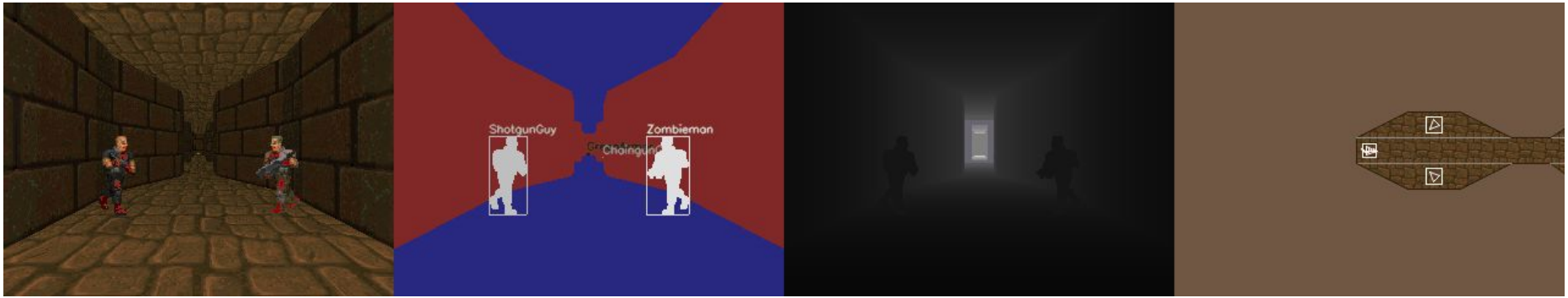


RandomAgent on SpaceInvaders-v0



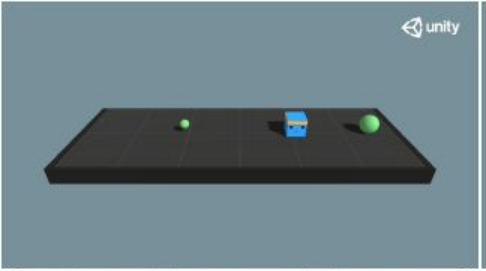
VIZDOOM allows developing AI bots that play **DOOM** using the visual information (the screen buffer). It is primarily intended for research in machine visual learning, and deep reinforcement learning, in particular.

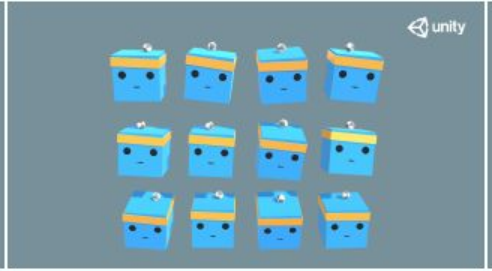
So far **VIZDOOM** was used as research platform in over 150 articles!

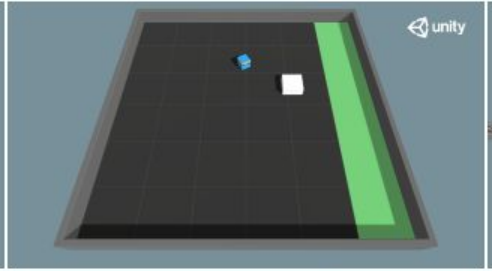


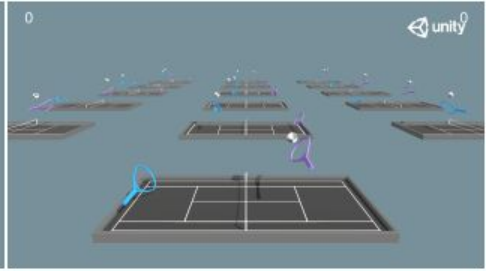

StarCraft II
2700
0
2
Remaining Time 0:01:32
Main Objectives
Collect Mineral Shards
Curriculum Score: 27
+100
F1
F2
4:37
Menu
M
S
H
P
A
T

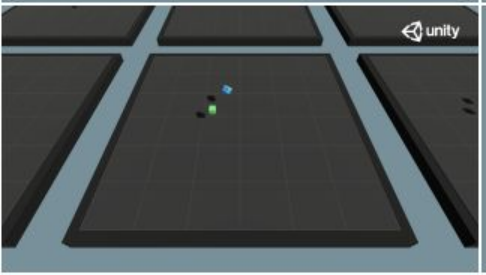

GRONINGEN

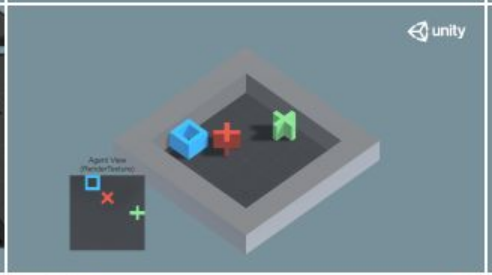

unity

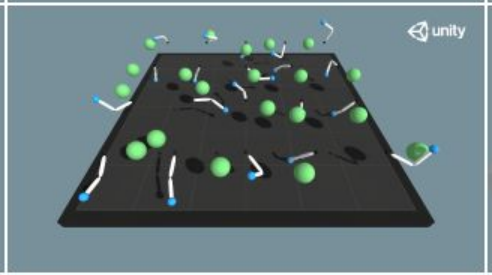

unity

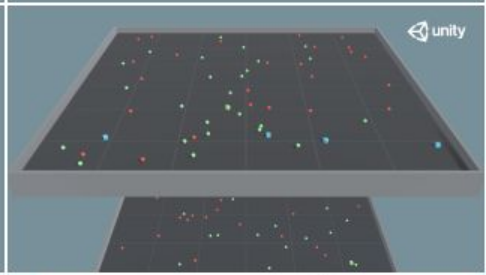

unity

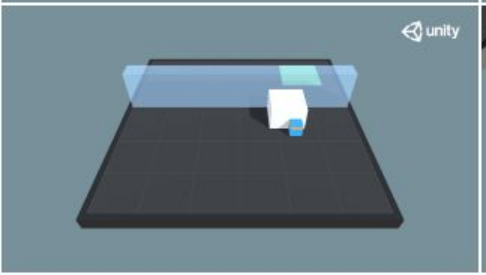

0
0
unity

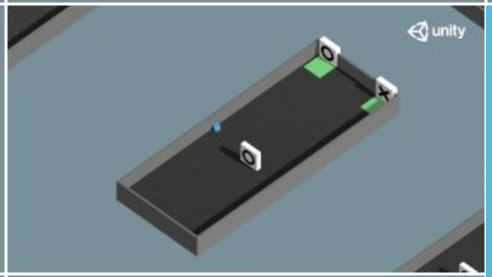

unity

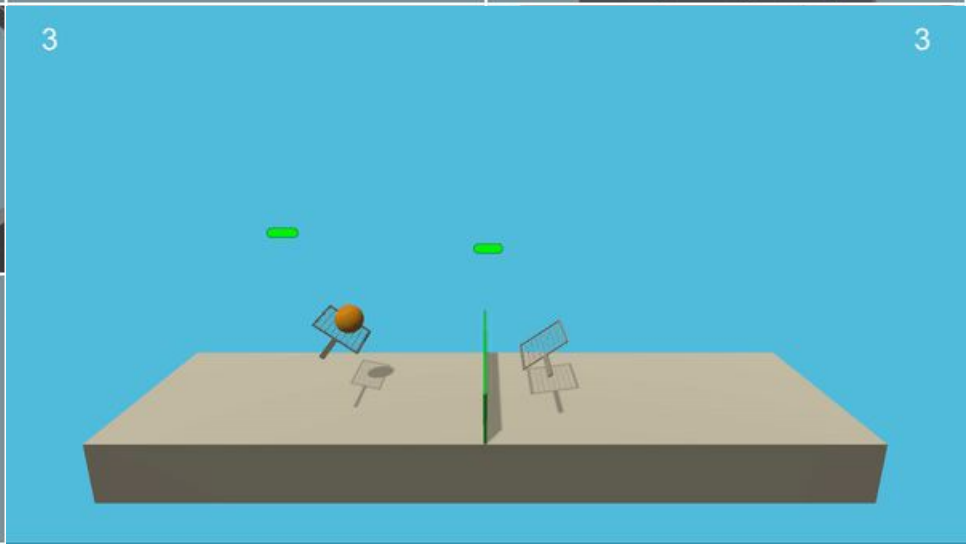

unity

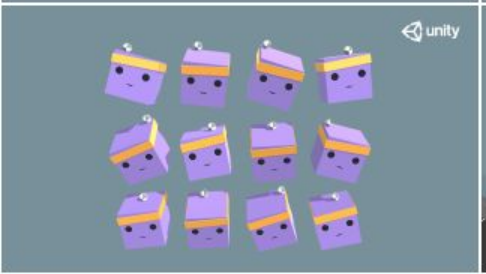

unity

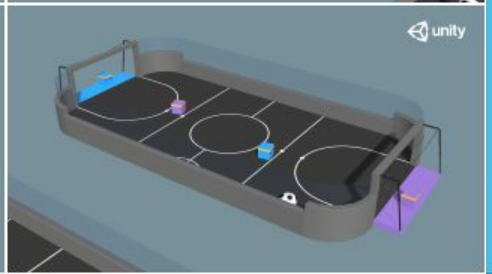

unity

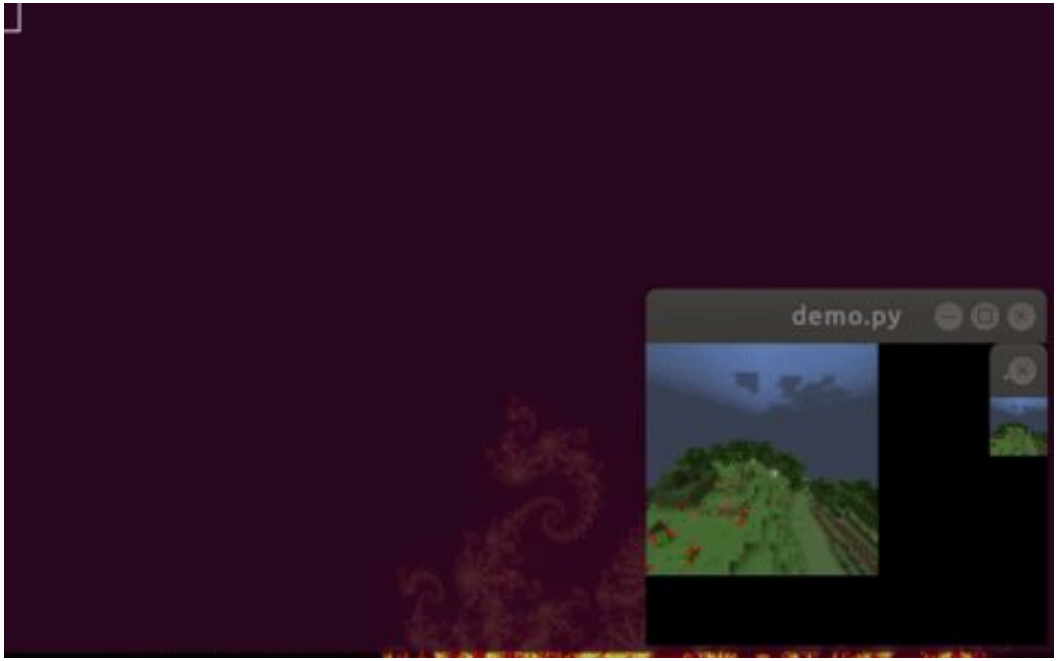

unity

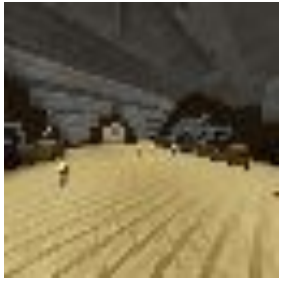

unity

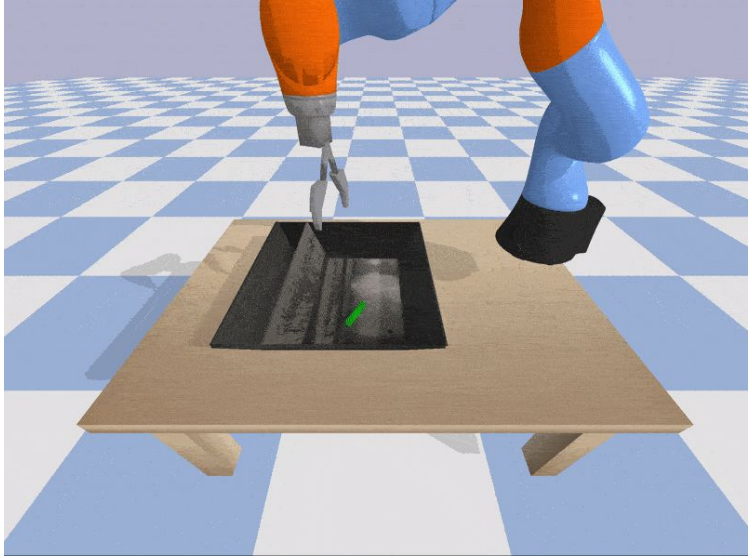

3
3

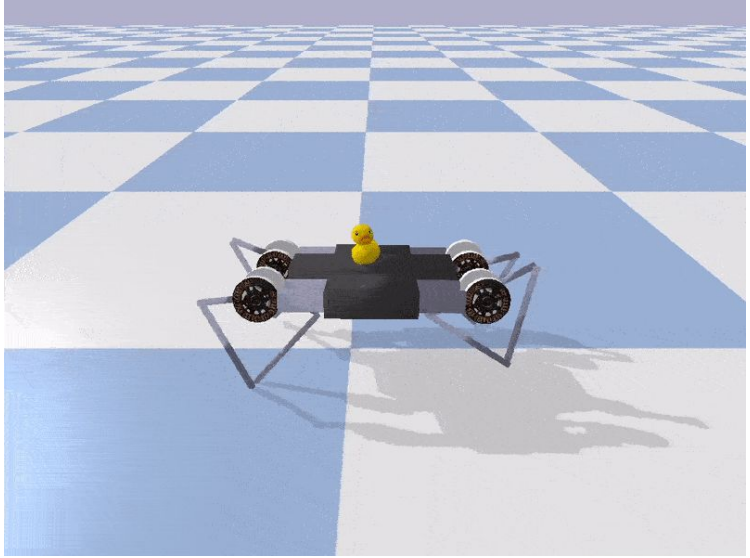

unity

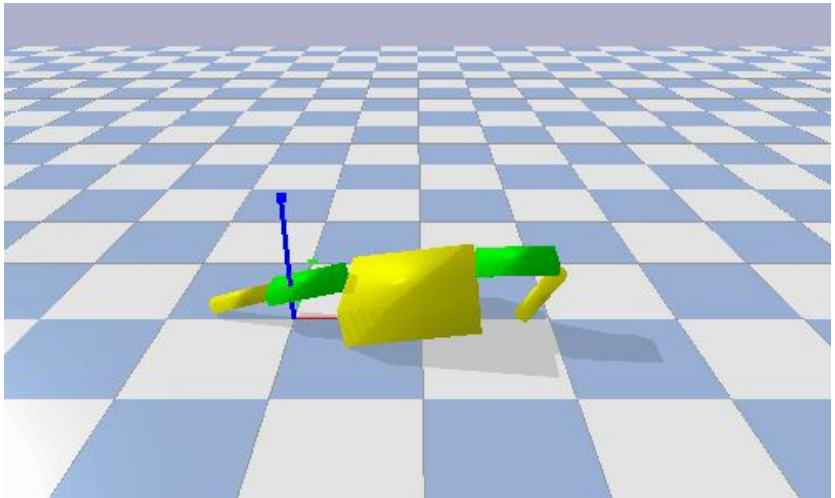

unity


demo.py









Visualization of Deep RL is a *hot* research area

Visual Diagnostics for Deep Reinforcement Learning Policy Development

Jieliang Luo*, Sam Green*, Peter Feghali, George Legrady, and Çetin Kaya Koç
University of California, Santa Barbara
jieliang@ucsb.edu

***Abstract*—Modern vision-based reinforcement learning techniques often use convolutional neural networks (CNN) as universal function approximators to choose which action to take for a given visual input. Until recently, CNNs have been treated like black-box functions, but this mindset is especially dangerous when used for control in safety-critical settings. In this paper, we present our extensions of CNN visualization algorithms to the domain of vision-based reinforcement learning. We use a simulated drone environment as an example scenario. These visualization algorithms are an important tool for behavior introspection and provide insight into the qualities and flaws of trained policies when interacting with the physical world. A video may be seen at https://sites.google.com/view/drlvisual.**

***Index Terms*—reinforcement learning, cyber-physical systems, convolutional neural networks, engineering visualization**

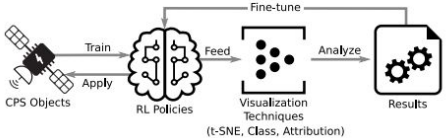
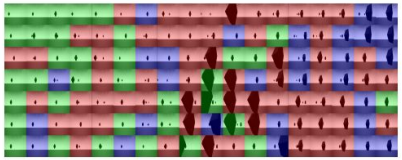
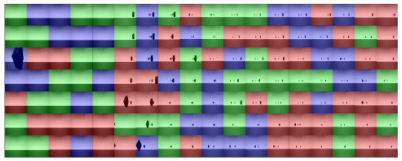


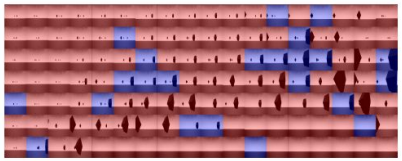
Fig. 1: Visualization development cycle when using convolutional neural networks for vision-based reinforcement learning. By iteratively visualizing *what* and *how* the CNN is perceiving, the engineer gains insight regarding *why* the RL policy makes its decisions.



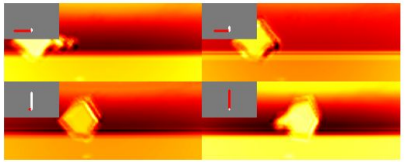
(a) High-performance policy



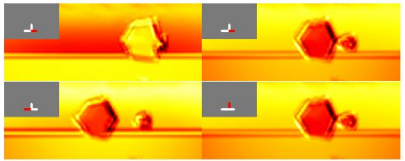
(b) Poor-performance policy



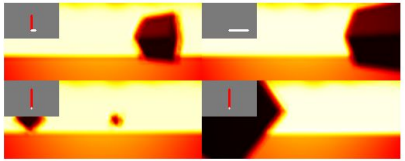
(c) Forward-and-right-only policy



(a) High-performance policy



(b) Poor-performance policy



(c) Forward-and-right-only policy

2018

DQNViz: A Visual Analytics Approach to Understand Deep Q-Networks

Junpeng Wang, Liang Gou, Han-Wei Shen, *Member, IEEE*, and Hao Yang

Fig. 1. *DQNViz*: (a) the *Statistics* view presents the overall training statistics with line charts and stacked area charts; (b) the *Epoch* view shows epoch-level statistics with pie charts and stacked bar charts; (c) the *Trajectory* view reveals the movement and reward patterns of the DQN agent in different episodes; (d) the *Segment* view reveals what the agent really sees from a selected segment.

2018

Visualizing and Understanding Atari Agents

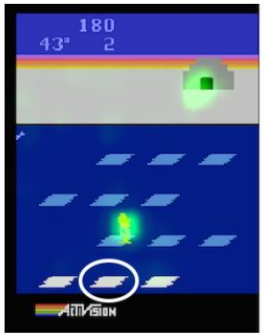
Sam Greydanus [1] Anurag Koul [1] Jonathan Dodge [1] Alan Fern [1]

Abstract

While deep reinforcement learning (deep RL) agents are effective at maximizing rewards, it is often unclear what strategies they use to do so. In this paper, we take a step toward explaining deep RL agents through a case study using Atari 2600 environments. In particular, we focus on using saliency maps to understand how an agent learns and executes a policy. We introduce a method for generating useful saliency maps and use it to show 1) what strong agents attend to, 2) whether agents well in challenging environments that have sparse rewards and noisy, high-dimensional inputs. Simply observing the policies of these agents is one way to understand them. However, explaining their decision-making process in more detail requires better tools.

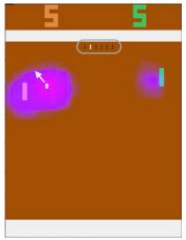
In this paper, we investigate deep RL agents that use raw visual input to make their decisions. In particular, we focus on exploring the utility of visual saliency to gain insight into the decisions made by these agents. To the best of our knowledge, there has not been a thorough investigation of saliency for this purpose. Thus, it is unclear which saliency methods

(a) MsPacman

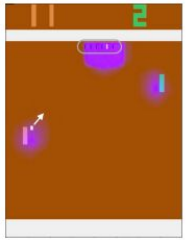


(b) Frostbite

(c) Enduro



(a) Pong: control



(b) Pong: overfit

(c) SpaceInvaders: control

(d) SpaceInvaders: overfit

(e) Breakout: control

(f) Breakout: overfit

2018

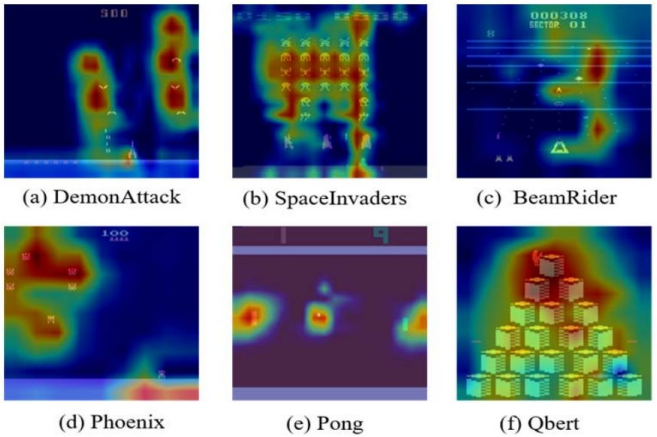
Visualization of Deep Reinforcement Learning using Grad-CAM: How AI Plays Atari Games?

Ho-Taek Joo
Institute of Integrated Technology
GIST
Gwangju, South Korea
ureca87@gmail.com

Kyung-Joong Kim
Institute of Integrated Technology
GIST
Gwangju, South Korea
kjkim@gist.ac.kr

***Abstract* — Deep Reinforcement Learning (DRL) allows agents to learn strategies to solve complex tasks. It has been applied to solve various problems such as natural language processing, games, etc. However, it is still difficult to apply DRL to certain real-world problems because each action is not predictable, and we cannot know why the results are coming out. For this reason, a technology called eXplainable Artificial Intelligence (XAI) has been recently developed. As this technology shows a visualization of the AI process, people can**

policy. A3C's biggest novelty is asynchronous structure to improve learning speed and performance by using a global network and multiple parallel agents. Each agent learns policy through interaction in its environment, calculating the gradient and asynchronously updates to the global network. Each agent can make different explorations because the experience of each agent is independent. It is faster, more robust, and can score better than DQN.



(a) DemonAttack (b) SpaceInvaders (c) BeamRider

(d) Phoenix (e) Pong (f) Qbert

2019

FINDING AND VISUALIZING WEAKNESSES OF DEEP REINFORCEMENT LEARNING AGENTS

Christian Rupprecht[1] **Cyril Ibrahim**[2] **Christopher J. Pal**[2,3]

[1] *Visual Geometry Group, University of Oxford*
[2] *Element AI*
[3] *Polytechnique Montréal, Mila & Canada CIFAR AI Chair*

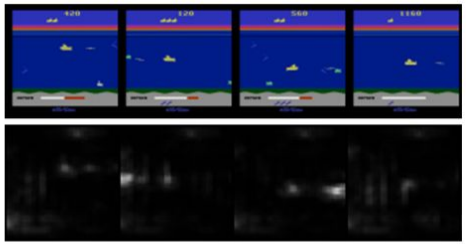


Figure 2: **Weight Visualization.** We visualize the weighting (second row) of the reconstruction loss from Equation 2 for eight randomly drawn samples (first row) of the dataset. Most weight lies on the player's submarine and close enemies, supporting their importance for the decision making.

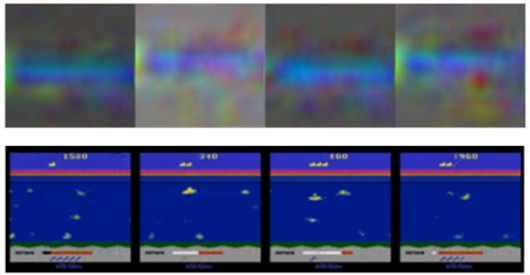


Figure 3: **Comparison with activation maximization.** The visual features learned by the agents are not complex enough to reconstruct typical frames from the game via activation maximization (top). This problem is mitigated in our method by learning a low-dimensional embedding of games states (bottom).

Thank you!

Paulo Bruno Serafim

paulobruno@alu.ufc.br
paulobruno.github.io