CS 2310 - Multimedia Software Engineering

Deep Reinforcement Learning

Mohammad H. Mofrad
University of Pittsburgh
Thursday, October 27, 2016

Preface

- Machine Learning
- Reinforcement Learning
- Artificial Neural Network
- Deep Learning
- Deep Reinforcement Learning

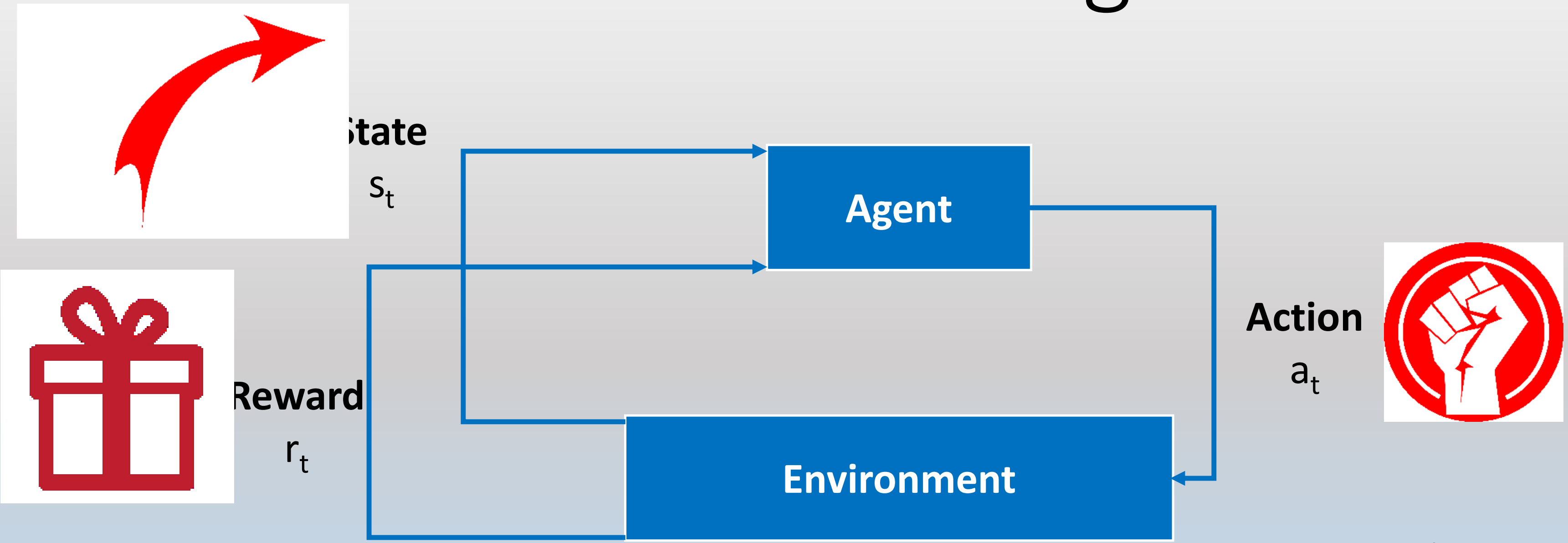
Machine Learning

- ML in a Nutshell

$y = f(x)$

- Supervised Learning
  - Discrete space: **X (classification) Y**
  - Continuous space **X (regression) Y**
- **Reinforcement Learning**
- Unsupervised Learning
  - Discrete space: X (clustering)Y
  - Continuous space: X (dimensionality reduction) Y

Reinforcement Learning



Reward

Action

State

Terminal State

$s_0, a_0, r_1, s_1, a_1, r_2, s_2, s_2, \ . \ . \ . \ , r_{n-1}, s_{n-1}, a_{n-1}, r_n, s_n$

https://webdocs.cs.ualberta.ca/~sutton/book/ebook/node28.html

Q – Learning

- Select an action
- Observe the reward
- Update the Q – table

Reward

Learned value

$$Q(s_t, \alpha_t) = Q(s_t, a_t) + \alpha \left(\overbrace{r_{t+1}\,\gamma\ \left(\underbrace{\max_{\alpha} \mathrm{Q}(s_{t+1}, \alpha)}_{\text{Estimate of optimal new state}}\right)}^{\text{Learned value}} - Q(s_t, \alpha_t) \right)$$

New state

Old state

Learning rate

Discount factor

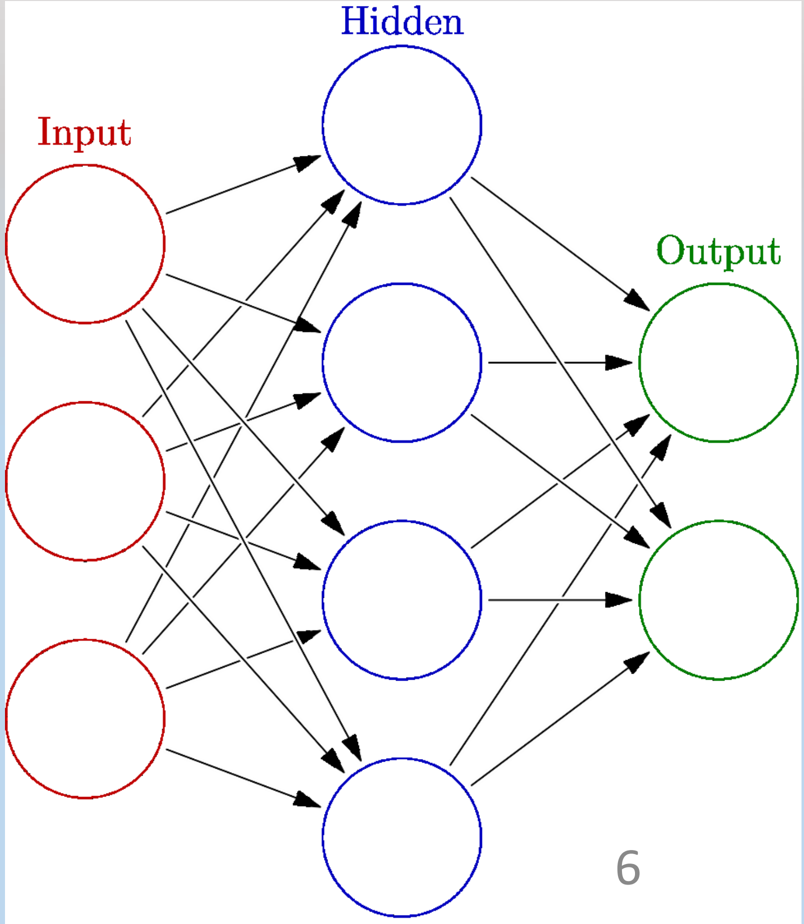
Estimate of optimal new state

Old state

https://en.wikipedia.org/wiki/Q-learning

Artificial Neural Network (ANN)

- A system of loosely coupled neural units modeling the brain neurons connected by axons.
- Mathematical model: **f: X → Y**
- Network structure:
  - f is the Neuron's network function: **$f(x) = K(\sum_i w_i g_i(x))$**
  - $x = (x_1, x_2, \ldots, x_n)$ is the input vector
  - $w = (w_1, w_2, \ldots, w_n)$is the weight vector
  - $g = (g_1, g_2, \ldots, g_n)$ is the composition of other functions
  - K is the activation function
- Learning: ANNs learn using a cost function
  - **$C = E[(f(x) - y)^2]$** like mean squared error
  - **$C' = 1/N \sum_i (f(x_i) - y_i)^2$** where N is the number of samples



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

Deep Learning

- A class ANN which is a combination of many layers of nonlinear processing units

+ Feature extraction

+ Classification

+ pattern recognition

- Combination of heterogeneous algorithms, mainly unsupervised
- Learn different levels of representation
- The output can be used as a feature vector for other classification schemes

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

Deep Reinforcement Learning

DeepMind

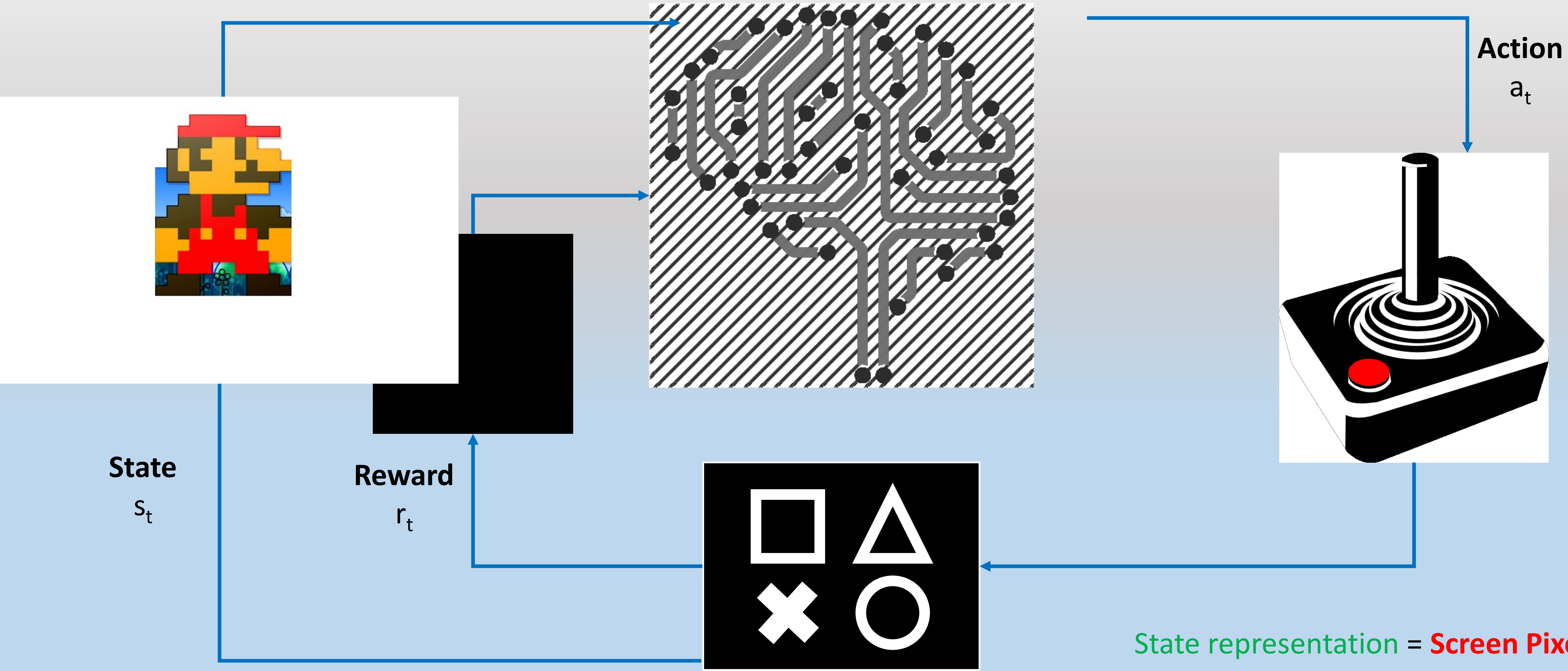
- British-based AI Company founded in 2010
- Acquired by Google in 2014
- A Neural Network that learns how to play video games
- Neural Turing Machine
- Healthcare
  - Searching for early signs of diseases leading to blindness.
  - Differentiate between healthy and cancerous tissues in head and neck area.
- AlphaGo

https://en.wikipedia.org/wiki/DeepMind
https://deepmind.com/research/alphago/

Selected Article

- Title: **Human-level control through deep reinforcement learning**
- Authors: Volodymyr Mnih et al.
- Affiliation: Google DeepMind
- Journal: **Nature** – International Weekly Journal of Science
- Volume: 518
- Issue: 7540
- Date: 2015
- Pages: 529 – 533
- Journal Impact Factor: **38.138**
- Citation: **542**

Reinforcement Learning in Atari



State representation = **Screen Pixels**

Deep Q – Learning (DQL)

- Deep Learning + Reinforcement Learning?
- Bellman Equation with Value Function Q(s, a)

$$Q(s_t, \alpha_t) = r + \gamma \ \left(\max_a Q(s_{t+1}, \alpha)\right)$$

- Where Value Iteration Algorithm can recursively find the optimal value
- Represent the Deep Q – Network’s Value Function by the with weights W

$$Q(s_t, \alpha, w) \approx Q(s_t, \alpha)$$

Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning."

Deep Q – Learning (DQL)

- Define a new Objective Function by Mean-Squared Error (MSE) in Q-values

$$L = E[(r + \gamma(\max_\alpha Q(s_{t+1}, \alpha, w)) - Q(st, at))^2]$$

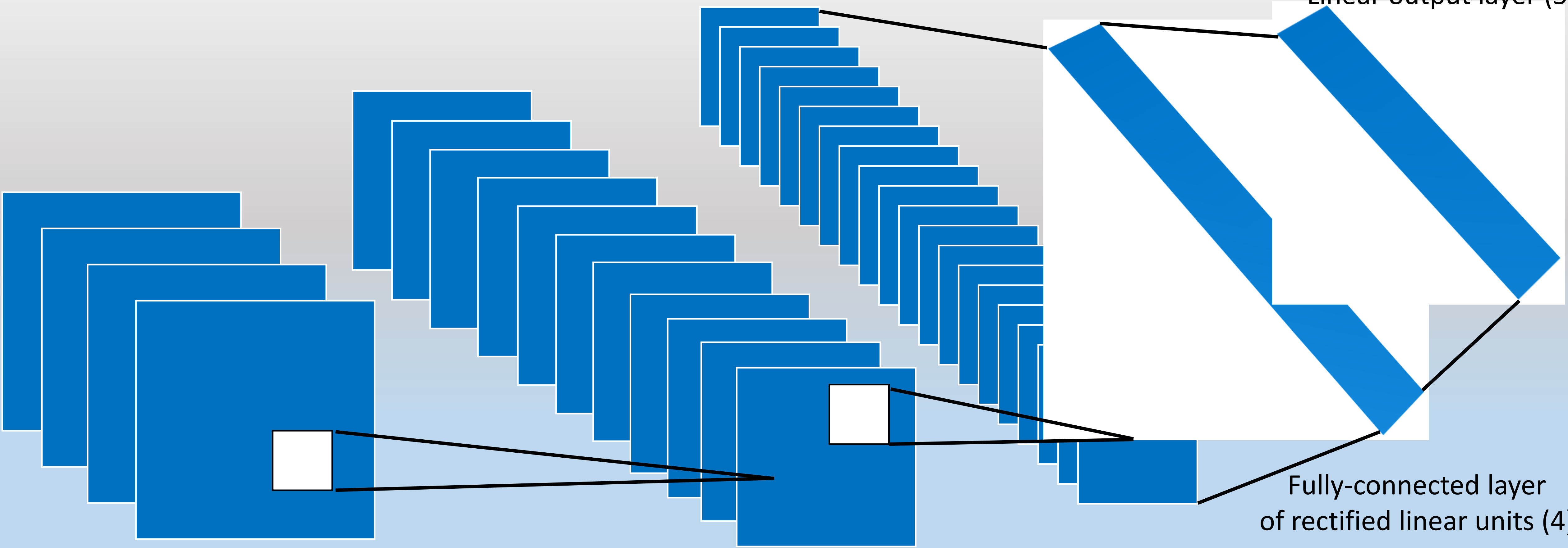
- Leading to the following Q – learning Gradient

$$\frac{\partial L(w)}{\partial L} = E\left[(r + \gamma(\max_\alpha Q(s_{t+1}, \alpha, w)) - Q(st, at, w))\frac{\partial Q(st, at, w)}{\partial w}\right]$$

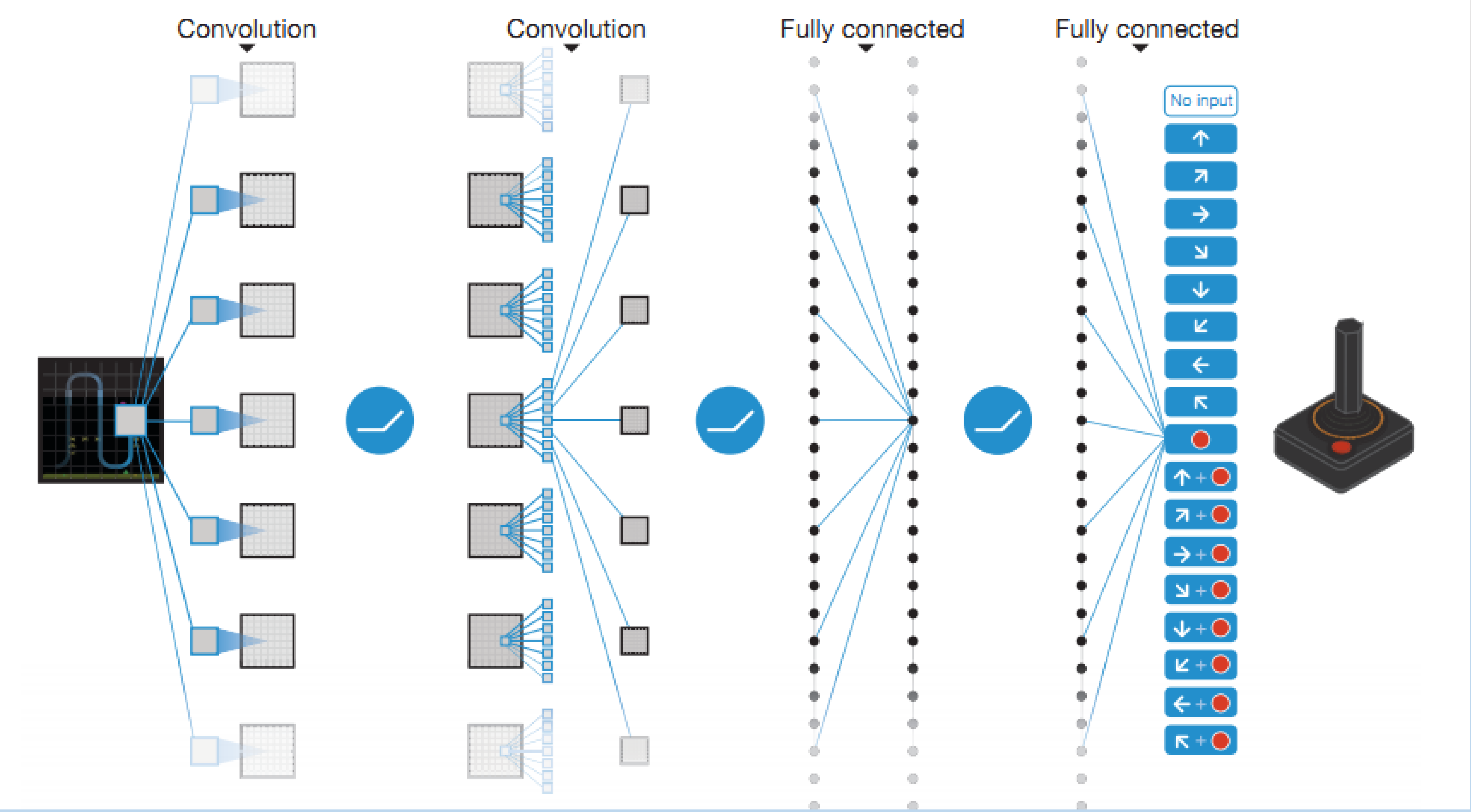
- Optimize objective function by Stochastic Gradient Descent (SGD) using $\frac{\partial L(w)}{\partial L}$

Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning."

The Convolutional Neural Network in Atari



Schematic illustration of Deep Q – Network



Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning."

Deep Q – Network characteristics

Layer	Input	Filter size	Stride	# Filters	Activation Function	Output
Convolution 1	84 x 84 x 4	8 x 8	4	32	ReLU	20 x 20 x 32
Convolution 2	20 x 20 x 32	4 x 4	2	64	ReLU	9 x 9 x 64
Convolution 3	9 x 9 x 64	3 x 3	1	64	ReLU	7 x 7 x 64
Fully connected 4	7 x 7 x 64			512	ReLU	512
Fully connected 5	512			18	Linear	18

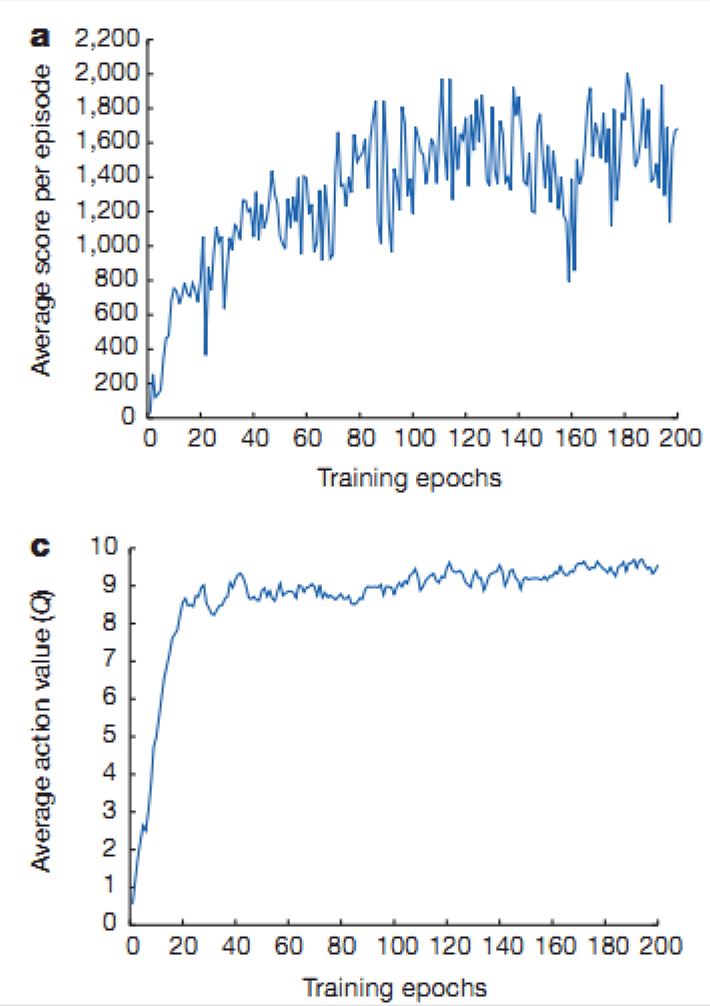
*Rectified Linear Unit (ReLU)

Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning."

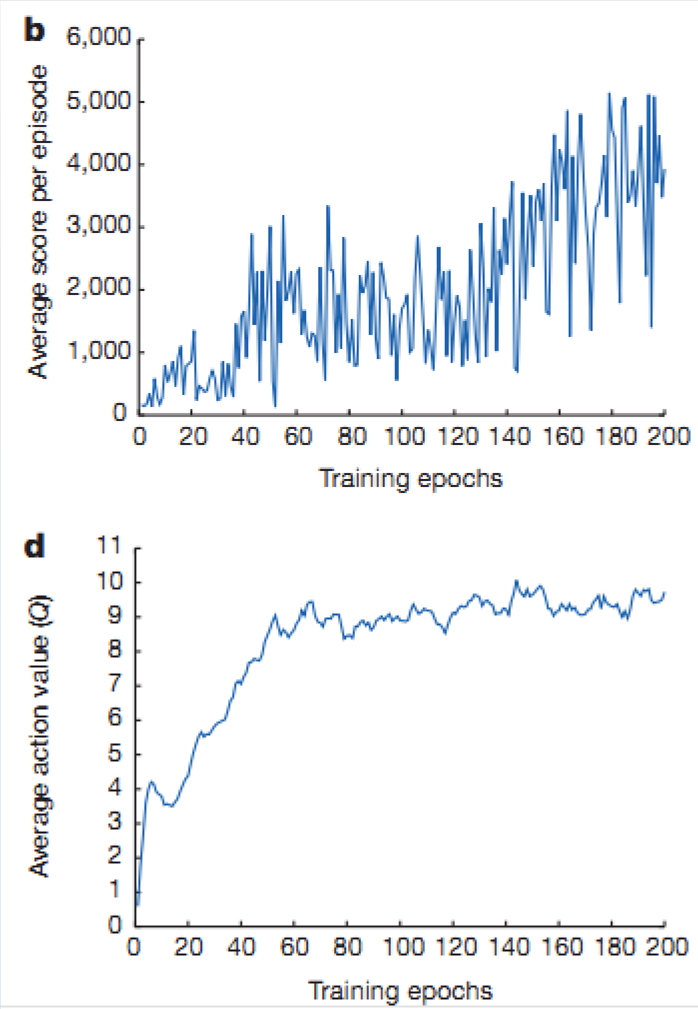
Results

Training Curve

Space Invaders



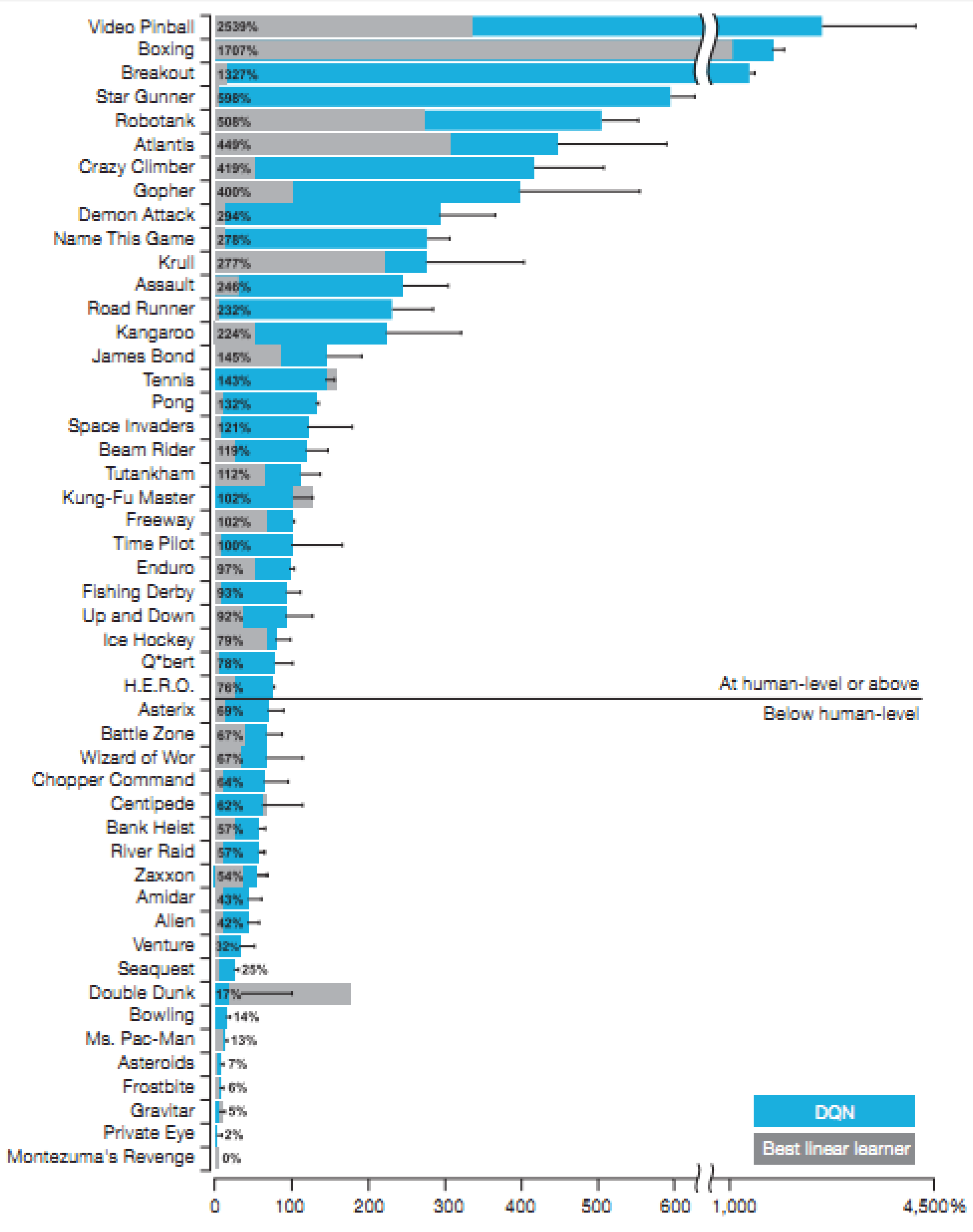
Seaquest



Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning."

Comparing DQN performance

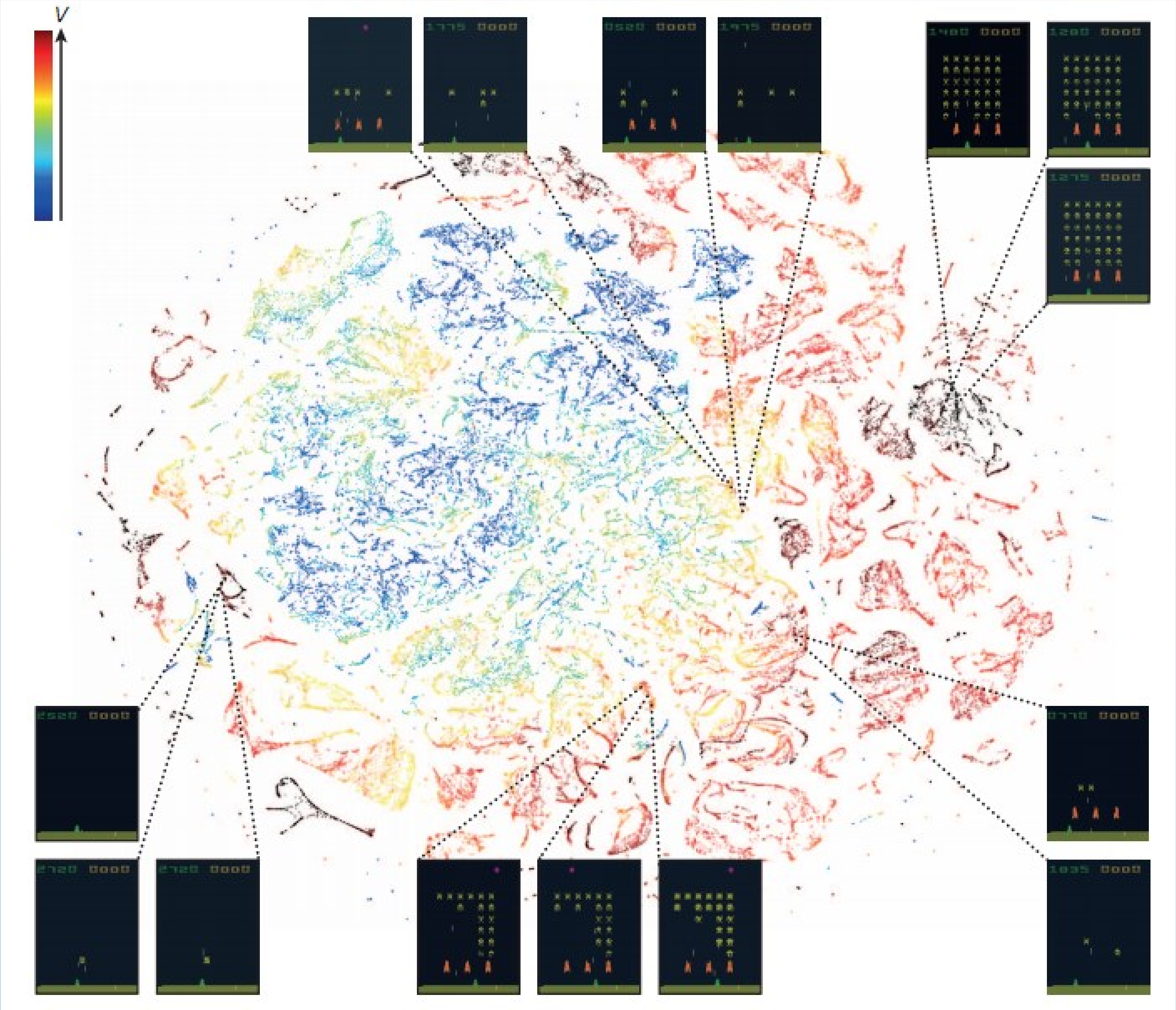
- Audio was disables
- 30 iterations
- Normalized results



Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning."

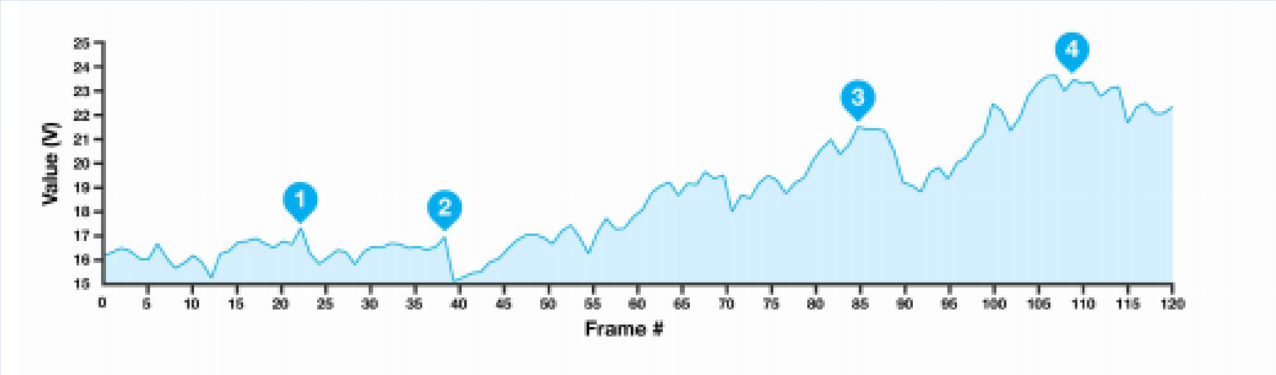
Last hidden layer representation

- tSNE representation of game states



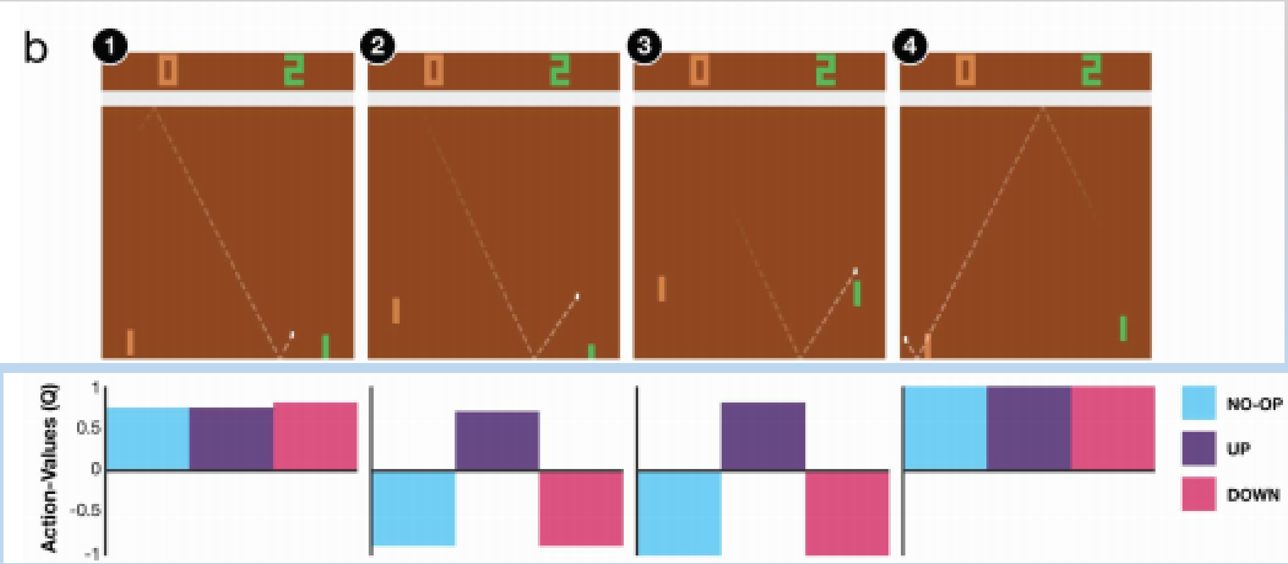
Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning."

Visualization of the learned value Function **Breakout** game



Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning."

Visualization of the learned value Function **Pong** game



Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning."

Conclusion

- Deep nets are most suitable while dealing with unlimited high dimensional training data
- DQN
  - Reinforcement Learning acts as the function approximator
  - Extract high level features from high dimensional raw sensory data
- Final Quote:

“Reinforcement learning + deep learning = AI”

References

[1] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." *Nature* 518.7540 (2015): 529-533.

[2] Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." *arXiv preprint arXiv:1602.01783* (2016).

[3] Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." *arXiv preprint arXiv:1312.5602* (2013).