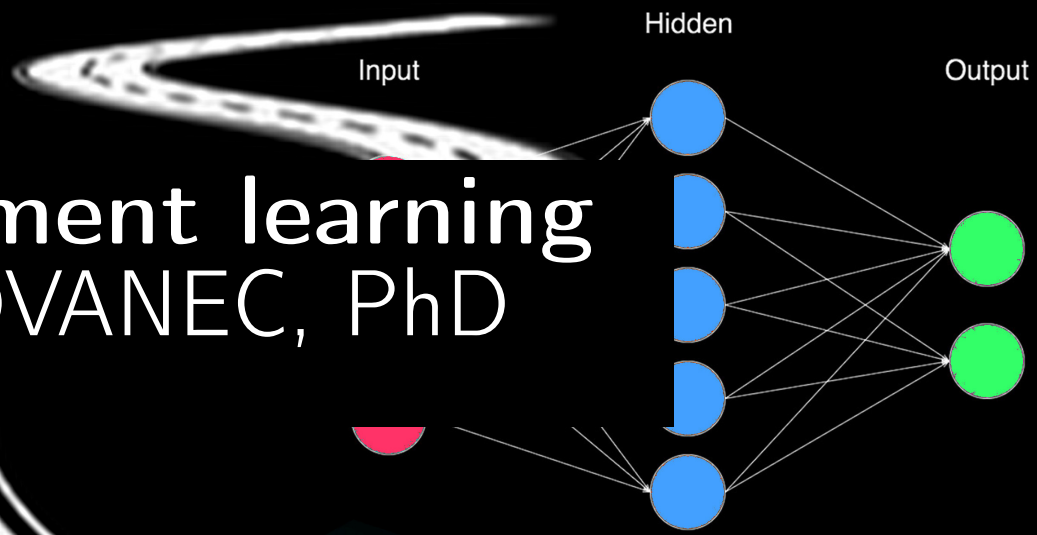
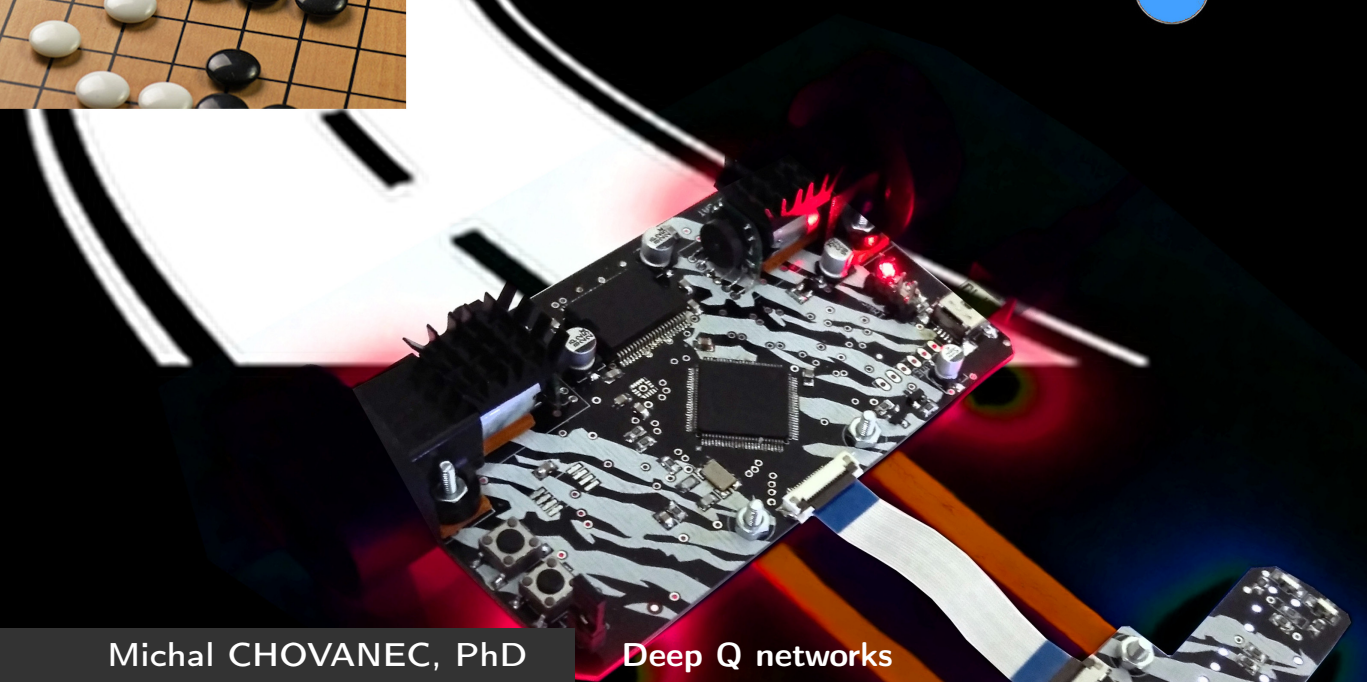
$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \lambda \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right]$$

(The New Action Value = The Old Value) + The Learning Rate × (The New Information − the Old Information)



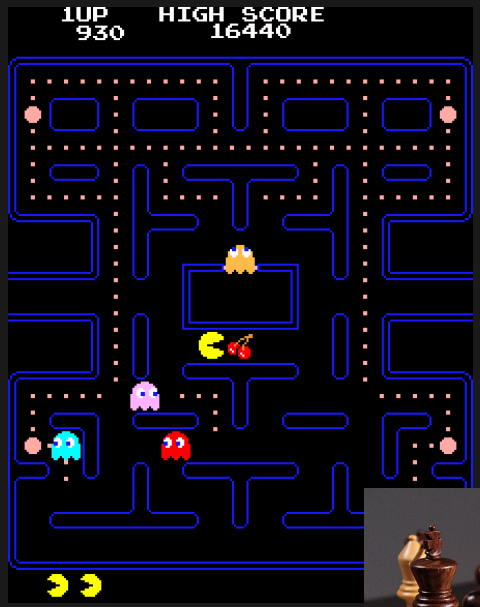
Reinforcement learning

Michal CHOVANEC, PhD



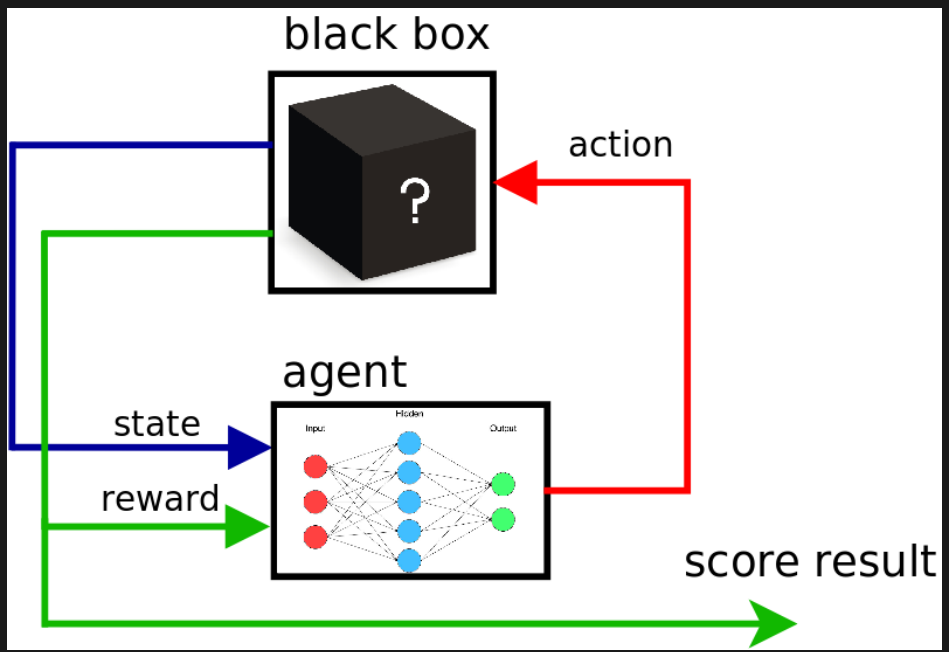
Reinforcement learning

- learn from punishment and rewards
- learn to play a game with unknown rules



Reinforcement learning

- obtain **state**
- choose **action**
- **execute** action
- obtain **reward**
- learn from **experiences**



Q learning

$$Q'(s,a) = R + \gamma \max_{\alpha'} Q(s', \alpha')$$

where
s is state
a is action
s' is next state
a' is best action in next state
$R(s,a)$ is reward
$\gamma \in \langle 0, 1 \rangle$ is discount factor

Deep Q learning

$$\hat{Q}(s, a; w) = R + \gamma \max_{\alpha} \hat{Q}(s', \alpha; w),$$
$$\hat{Q}(s, a; w) = R + \gamma \max_{\alpha} \hat{Q}(s', \alpha; w')$$

Usefull links


Andrej Karpathy : Pong from pixels
http://karpathy.github.io/2016/05/31/rl/


Richard S. Sutton : Reinforcement Learning: An Introduction
https://www.amazon.com/Reinforcement-Learning-Introduction-Adaptive-Computation/dp/0262193981


Maxim Lapan : Deep reinforcement learning
https://www.amazon.com/Practical-Reinforcement-Learning-Maxim-Lapan/dp/1788834240


Mohit Sewak : Practical Convolutional Neural Networks
https://www.amazon.com/Practical-Convolutional-Neural-Networks-Implement/dp/1788392302


CHRISTOPHER J.C.H. WATKINS : Q-learning
http://www.gatsby.ucl.ac.uk/~dayan/papers/cjch.pdf


Densely Connected Convolutional Networks
https://arxiv.org/pdf/1608.06993.pdf


Mastering the Game of Go without Human Knowledge
https://deepmind.com/documents/119/agz_unformatted_nature.pdf

Q&A

michal chovanec (michal.nand@gmail.com)
www.youtube.com/channel/UCzVvP2ou8v3afNiVrPAHQGg
github https://github.com/michalnand