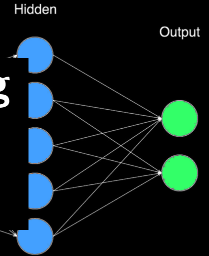
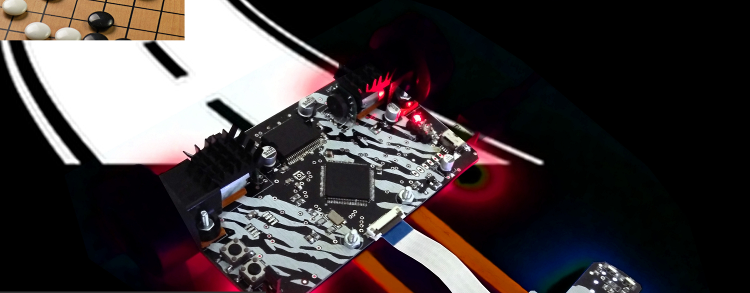
$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \lambda \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right]$$

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



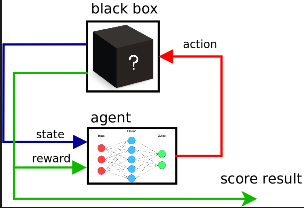
Reinforcement learning experiments

Michal CHOVANEC, PhD

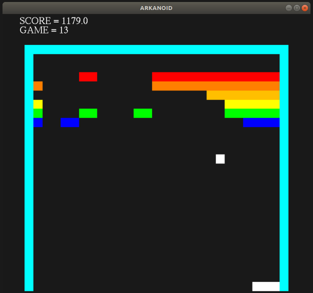


Reinforcement learning

- obtain **state**
- choose **action**
- **execute** action
- obtain **reward**
- learn from **experiences**
- function $Q(s, a)$, how good is action a in state s



- **playing Atari**
- **playing Doom**
- **playing GO**

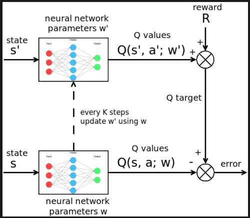


Deep Q network

- **correlated states** : experience replay buffer
- **unstable training** : non-stationary target value $\hat{Q}(s, a; w)$, depends on w, use temporary fixed weights w'
- **unknow gradients values** : clip or normalise rewards, Q values and gradients into $\langle -1, 1 \rangle$

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

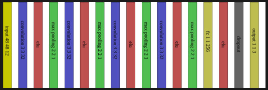
$$\mathcal{L} = (R + \gamma \max_{\alpha'} \hat{Q}(s', \alpha'; w') - \hat{Q}(s, a; w))^2$$



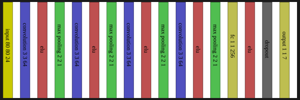
Networks architecture

Following modern **State of the art** networks :
3x3 convolutions, 2x2 pooling, ELU activation

- Atari
  - input : 48x48x12 (rgb x 4 last frames)
  - network : C3x3x32 - P2x2 - C3x3x32 - P2x2 - C3x3x32 - P2x2 - C3x3x32 - P2x2 - FC256 - $FC_{actions\_count}$



- DOOM
  - input : 80x80x24 (rgb x 8 last frames)
  - network : C3x3x64 - P2x2 - C3x3x64 - P2x2 - C3x3x64 - P2x2 - C3x3x64 - P2x2 - FC256 - $FC_{actions\_count}$



GO Network architecture

we need to go much deeper for GO

- **28, 35 layers**
  dense blocks + feature pooling layer
- **input**
  4 matrices 19x19: black stones, white stones, empty fields, active player
- **output**
  recommended moves 19x19 + 1 for pass = 362 outputs