

## Abstract

- First DL model to learn control policies directly from high-D sensory input using RL.
  - CNN trained with variant of Q-learning.
  - Atari 2600 computer games.
  - Only raw pixels are input.
- 

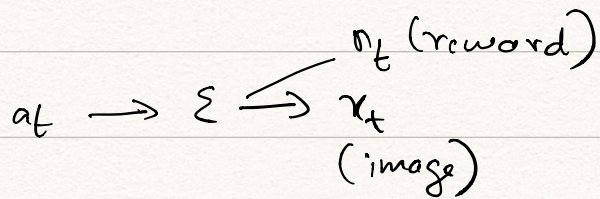
## 1) Introduction.

- DL generally require large hand-labelled data.
  - RL algorithm should learn from scalar reward signals that are largely sparse/noised/delayed.
  - DL requires data samples to be independent. RL have correlated data.
  - In RL, distribution changes as the algorithm learns new behaviors. DL assumes a fixed distribution.
  - Tested on Atari 2600 - (210x160 RGB video at 60Hz)
  - Goal is to create single NN agent that learns to play as many games as possible.
- 

## 2) Background.

$A = \{1 \dots K\}$  actions  $a_t$ , environment  $\Sigma$   
↓  
maybe stochastic





$s_t = \{x_1, a_1, x_2, \dots, a_{t-1}, x_t\}$  a sequence is a distinct state.  
 this sequence is assumed to terminate

$$R_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'} \quad (\text{goes from } t \text{ to completion})$$

$$Q^*(s, a) = \max_{\pi} E[R_t \mid s_t = s, a_t = a, \pi]$$

↓

optimal action value function

$$y_i = E[r + \gamma \max_{a'} Q_{\theta_{i-1}}(s', a')] \quad - \text{target}$$

$$L_i(\theta) = E[(y_i - Q_{\theta_i}(s, a))^2]$$

↓ the parameters from the previous iteration is held fixed while optimising the loss function.

$$\nabla L_i(\theta_i) = E[(\underbrace{r + \gamma \max_{a'} Q_{\theta_{i-1}}(s', a')}_{\text{target}} - \underbrace{Q_{\theta_i}(s, a)}_{\text{policy}}) \nabla_{\theta_i} Q_{\theta_i}(s, a)]$$

Model-free: directly samples from  $\mathcal{E}$ , without explicitly estimating  $\mathcal{E}$ .



opt. policy: learns greedy strategy  $\max_a Q(s, a)$   
with  $\epsilon$ -greedy exploration.

---

### 3) Related work

→ TD-gammon - model free RL - Q-learning with 1 hidden layer NN. Later chess / Go / checkers using the same approach was unsuccessful - (maybe it worked for backgammon because the stochasticity induced by dice rolls helped in exploration and made the value function smoother.)

→ Non-linear function approximator + model-free (Q-learning)  $\Rightarrow$  Q-network diverged.  $\Rightarrow$  majority of the work focussed on linear-fn-approximators.

→ Neural Fitted Q (NFQ) - solo author paper

(NFQ vs DQL?)

### 4) Deep Reinforcement Learning

- Goal: connect RL algorithm to a DNN which operates directly on images.

- Experience-replay - single author paper

At each time step  $e_t = (s_t, a_t, r_t, s_{t+1})$

$\downarrow$   
 $D = [e_1, e_2, \dots, e_N]$   
pooled over many episodes