Universidad Católica San Pablo
Departamento de Ciencia de la Computación
Centro de Investigación e Innovación en Ciencia de la Computación

Performing Deep Recurrent Double Q-Learning for Atari Games

Felipe Moreno-Vera

FONDECYT
FONDO NACIONAL DE DESARROLLO CIENTÍFICO, TECNOLÓGICO Y DE INNOVACIÓN TECNOLÓGICA
ProCIENCIA
CONCYTEC
CONSEJO NACIONAL DE CIENCIA, TECNOLOGÍA E INNOVACIÓN TECNOLÓGICA

Content

- Motivation
- Dataset
- Experiments
- Conclusions

Motivation

Deep Q-Learning

Algorithm 1 Deep Q-learning with Experience Replay

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ according to equation $\boxed{3}$
  end for
end for

This approach is in some respects limited since the memory buffer does not differentiate important transitions and always overwrites with recent transitions due to the finite memory size N.

Deep Double Q-Learning (DDQN)

DQN Model:

$$Y_t = R_{t+1} + \gamma max Q(S_{t+1}; a_t; \theta_t)$$

DDQN Model:

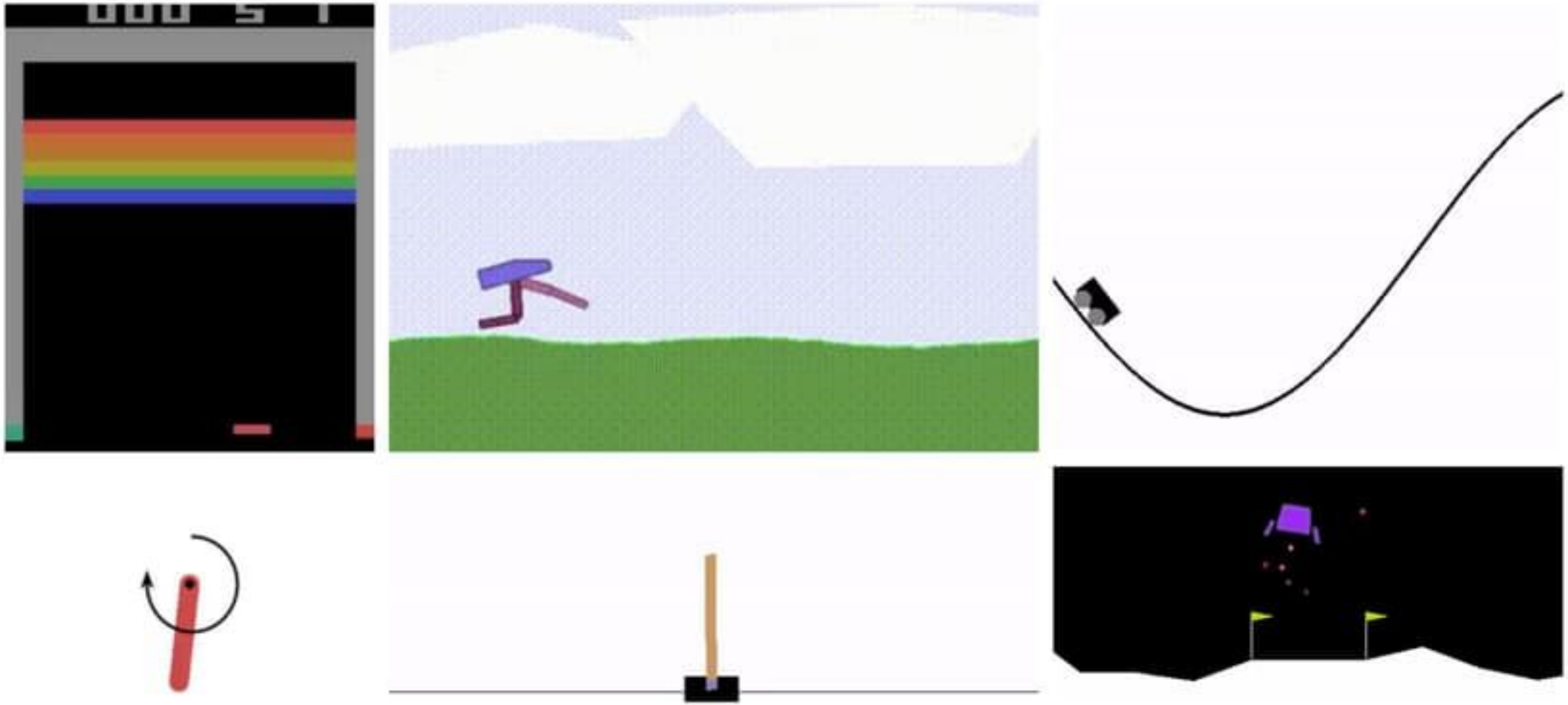
$$Y_t = R_{t+1} + \gamma Q(S_{t+1}; argmax Q(S_{t+1}; a_t; \theta_t); \theta_t^1)$$

Where:

- • a_t represents the agent.
- • θ_t are the parameters of the network.
- • Q is the vector of action values.
- • Y_t is the target updated resembles stochastic gradient descent.
- • γ is the discount factor that trades off the importance of immediate and later rewards.
- • S_t is the vector of states.
- • R_{t+1} is the reward obtained after each action.

Dataset

Atari Games



Why Atari Games?

Atari 2600 games (from the Arcade Learning Environment) are a **benchmark suite** in RL because they offer:

- High-dimensional **pixel-based inputs** (visual challenges).
- **Sparse and delayed rewards**—perfect for testing learning efficiency.
- A wide range of game mechanics, such as:
  - Reaction time (e.g., Pong)
  - Strategic planning (e.g., Breakout)
  - Exploration vs. exploitation (e.g., Montezuma’s Revenge)

Atari Games

OpenAI Gym is a popular toolkit for developing and comparing reinforcement learning algorithms. It provides:

- A **standardized interface** for a wide range of environments.
- **Benchmarking** tools for RL research.
- **Lightweight simulations** with easy-to-use APIs for training agents.

Gym is designed to support multiple types of environments, such as:

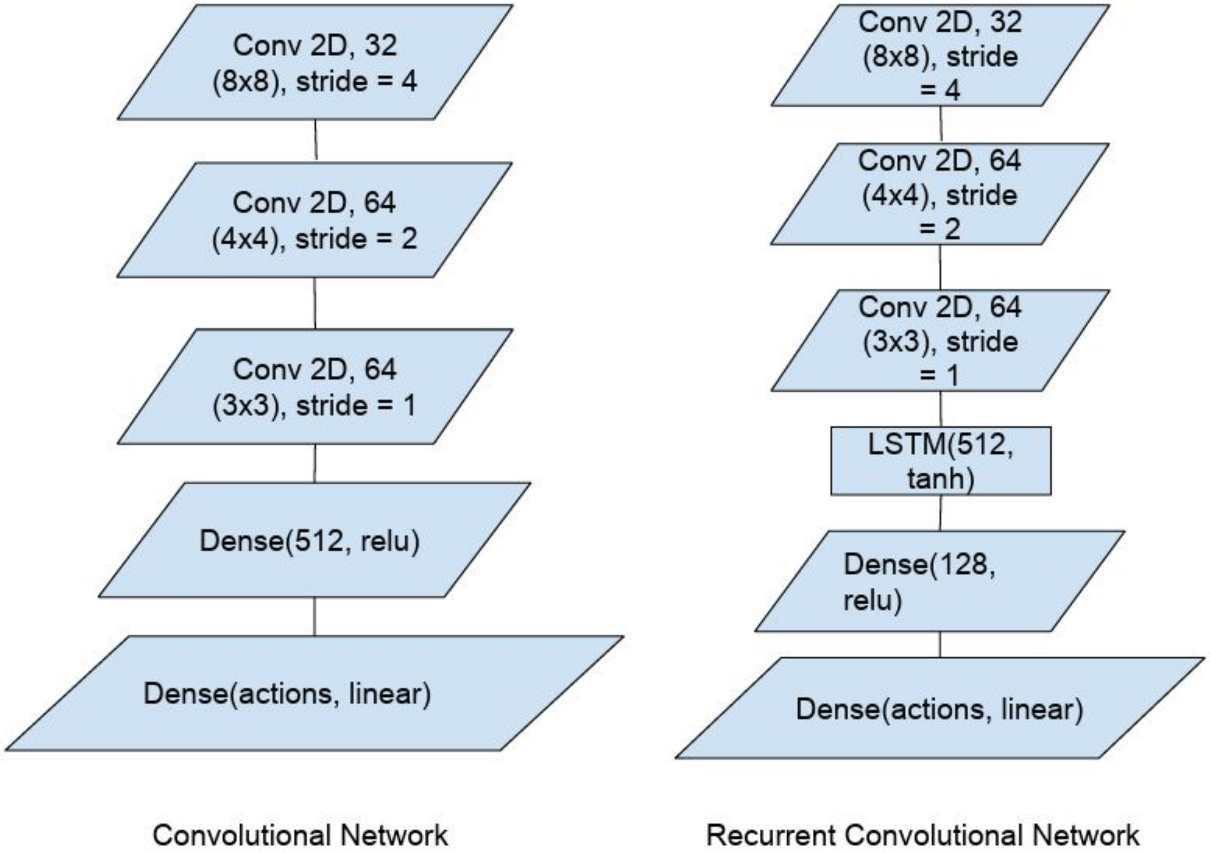
- Classic control (e.g., CartPole, MountainCar)
- Box2D physics-based tasks (e.g., LunarLander)
- Robotics simulations
- **Atari games** (via the Atari environment)

Experiments

Hyperparameters

List of Hyperparameters		
Iterations	10 000000	number of batch iterations to the learning process
miniBatch size	32	number of experiences for SGD update
Memory buffer size	900000	SGD update are sampled from this number of most recent frames
Learning Rate	0.00025	learning rate used by RMS Propagation
Training Frequency	4	Repeat each action selected by the agent this many times
Y Update Frequency	40000	number of parameter updates after which the target network updates
Update Frequency	10000	number of actions by agent between successive SGD updates
Replay start size	50000	The number of Replay Memory in experience
Exploration max	1.0	Max value in exploration
Exploration min	0.1	Min value in exploration
Exploration Steps	850000	The number of frames over which the initial value of e reaches final value
Discount Factor	0.99	Discount factor γ used in the Q-learning update

Our proposed model

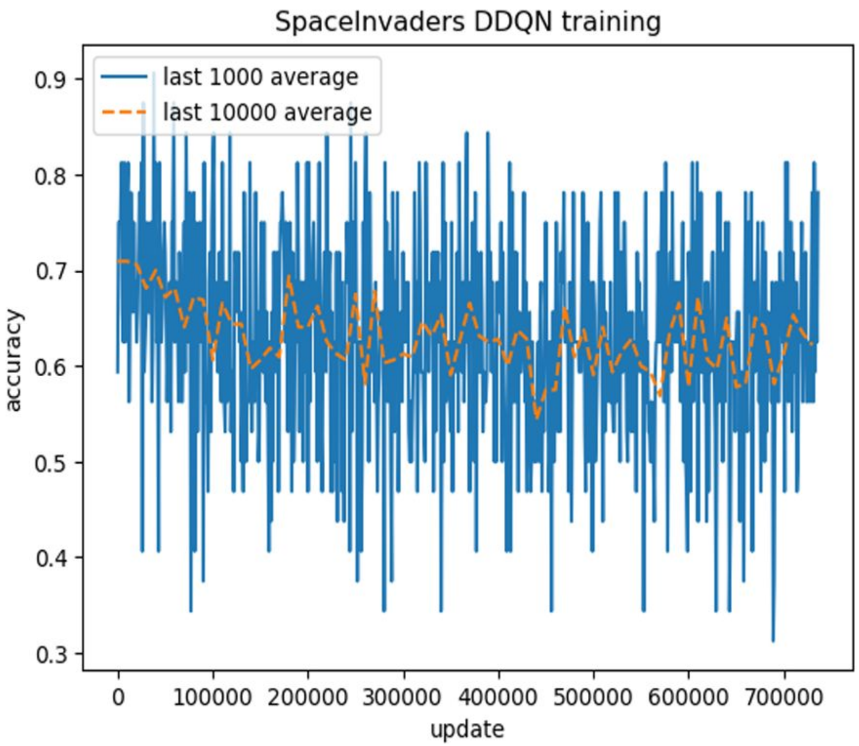


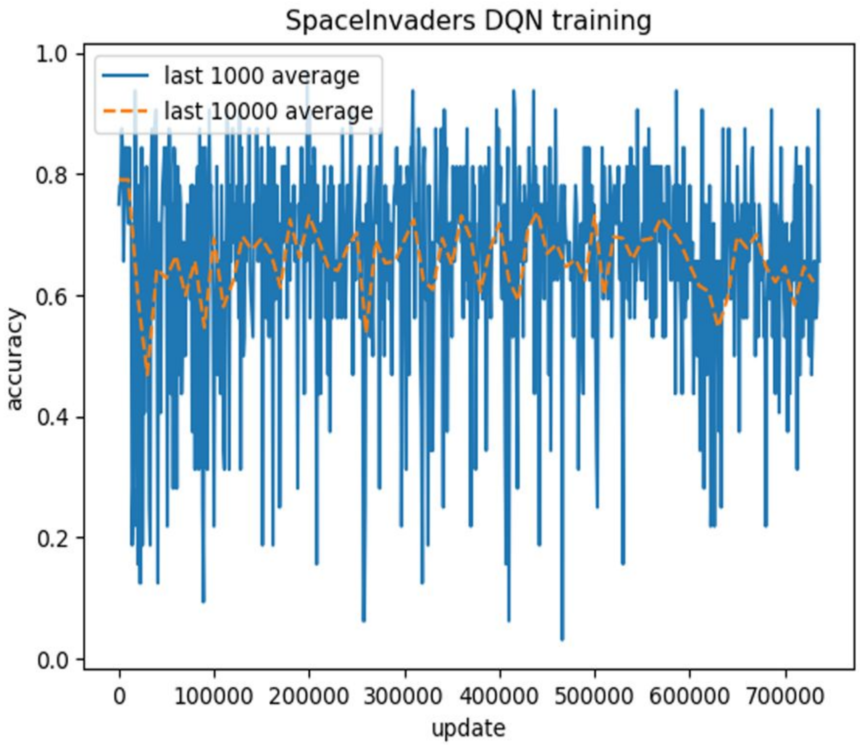
Model benchmarks

RESULTS SCORES OF SPACE INVADERS, ENDURO, PONG AND BEAM RIDER.

Models and respective Scores				
Model	SpaceInvaders	Enduro	Pong	Beam Rider
DQN	1450	1095	65	349
DRQN	1680	885	39	594
DDQN	2230	1283	44	167
DRDQN	2450	1698	74	876

Model history training





Conclusions

Conclusions

- We were able to analyze **Atari games using Reinforcement Learning**

- It is feasible to **train** an anget to **infer** the best **positions and rewards** in **Space invader game.**

- **Black-box CNN models** help to classify images.
    - It performs better than for image-based games.
    - We won't be able to perform a deep benchmark, due to limited computational resources.

Thanks! Any questions?

felipe.moreno@ucsp.edu.pe

THANKS!