Deep Q-Learning applied to air tactical deconfliction

CS231N – Spring 2022

Marta Palomar Toledano
mpalomar@stanford.edu

CS231N Project: Deep Q-Learning applied to air tactical deconfliction

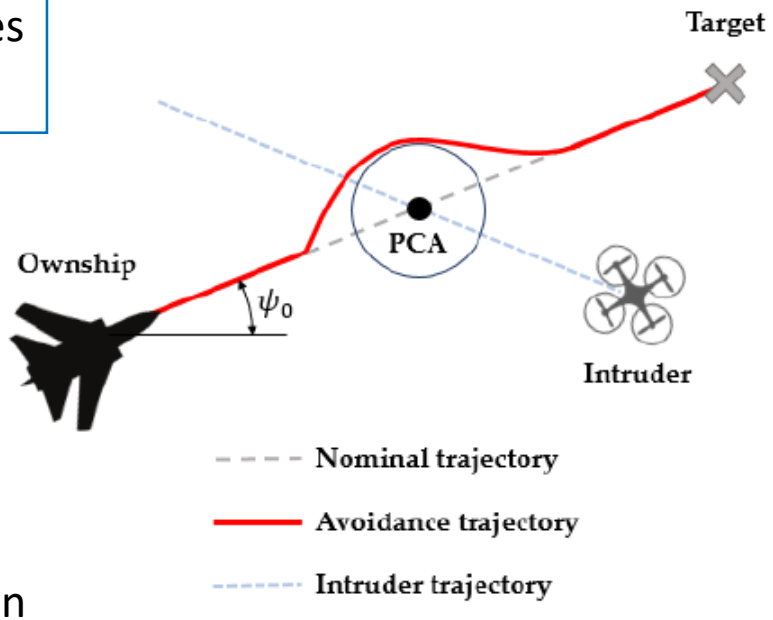
Introduction and Problem Statement

Purpose

Create a tool for Conflict Detection and Resolution (CDR) between flying vehicles based on their trajectories using AI.

Problem statement

One aircraft (ownship) is following its nominal trajectory towards a target and one (or more) intruders appear presenting a potential collision conflict.
Goal: apply a series of avoidance maneuvers until all conflicts are solved and retake nominal trajectory to the target.



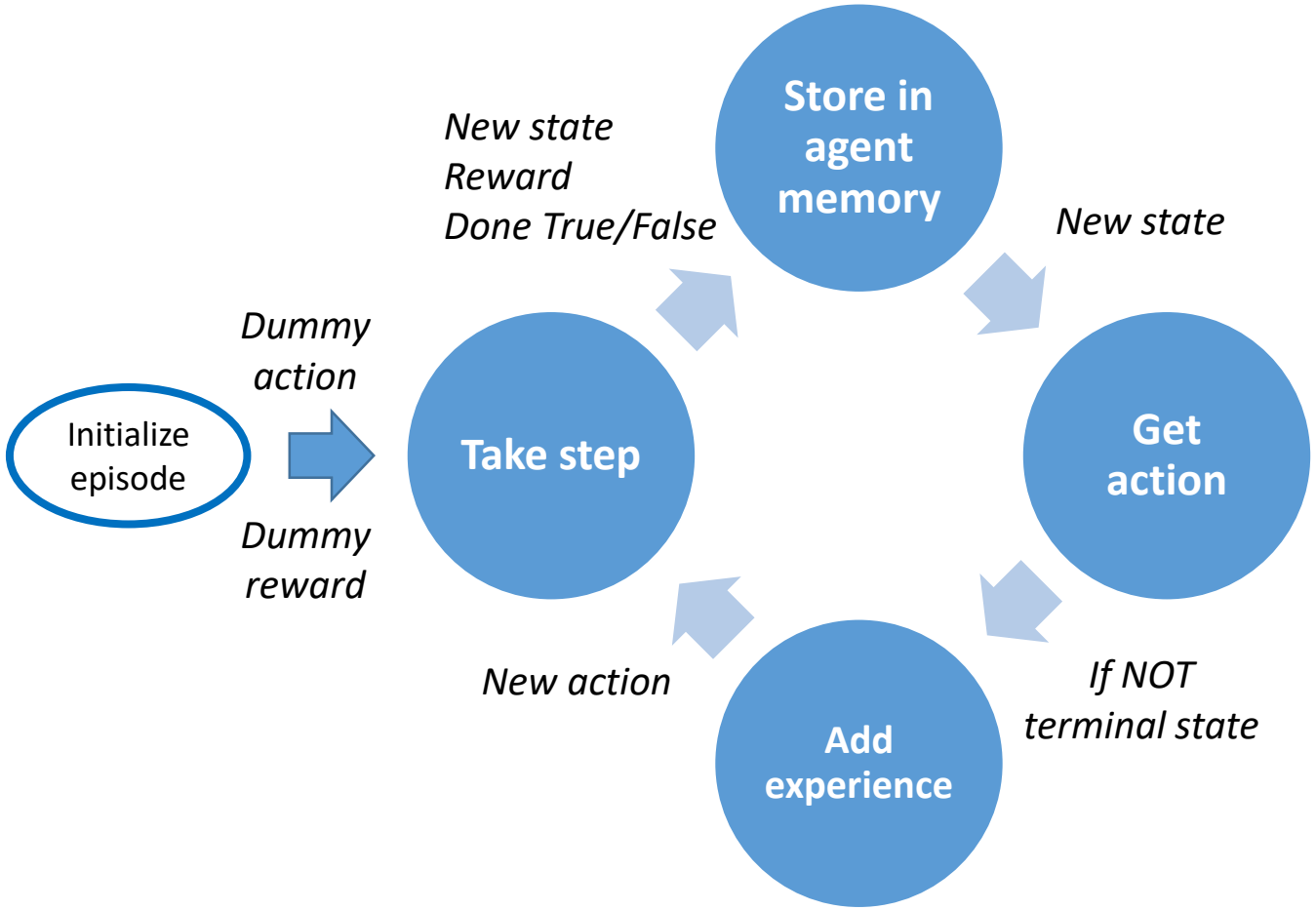
- Relevant due to growth in the volume of air operations due to improvement in autonomous technologies (Unmanned Air Vehicles).

CS231N Project: Deep Q-Learning applied to air tactical deconfliction

Methodology

Main methods employed: CNN + DQN arquitecture to create an agent.

- Input data (points from a trajectory) converted to image format. Idea from literature:
  - By using images, the model can easily adapt to a large number of intruders without altering state space.
  - Take advantage of CNNs (data augmentation, etc)
- Input data are synthetically generated due to lack of real ATC data from air conflicts.
- OpenAI Gym custom environment + Tensorflow backend.

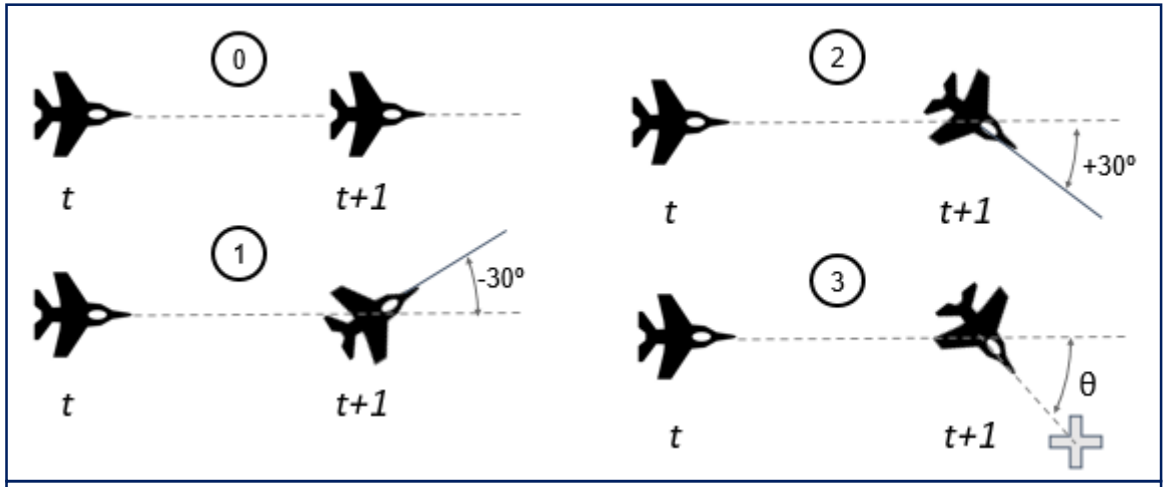


CS231N Project: Deep Q-Learning applied to air tactical deconfliction

Learning workflow

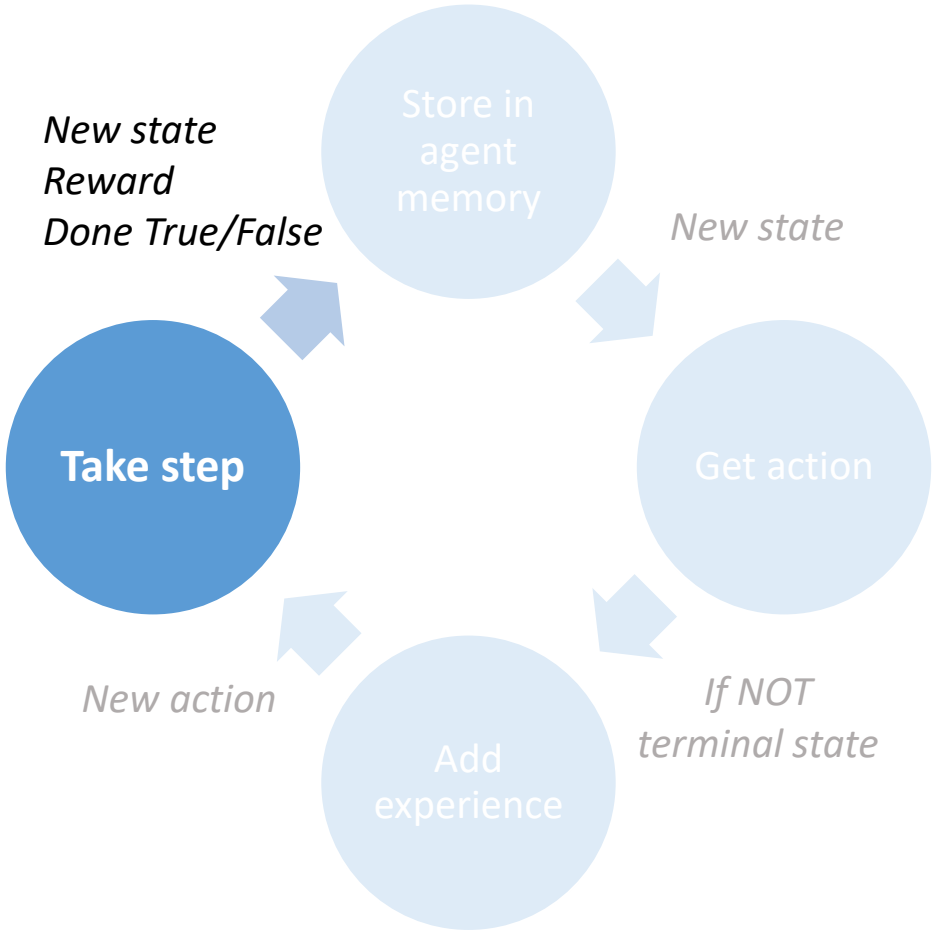
- ***Take step***: given an action from the available action space, the aircraft moves accordingly and a certain reward is received.

Action Space



Rewards

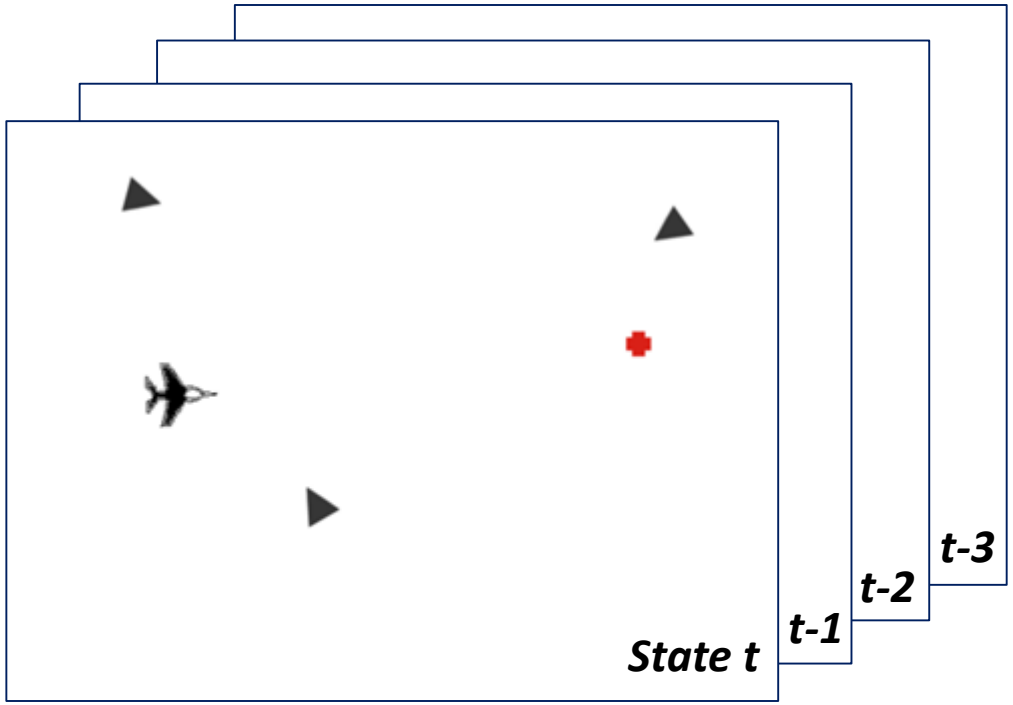
$$r(a) = \begin{cases} 0 & d_{intruder} > d_{critical} \\ -\dfrac{d_{critical} - d_{intruder}}{d_{critical}} & d_{intruder} \leq d_{critical} \end{cases}$$

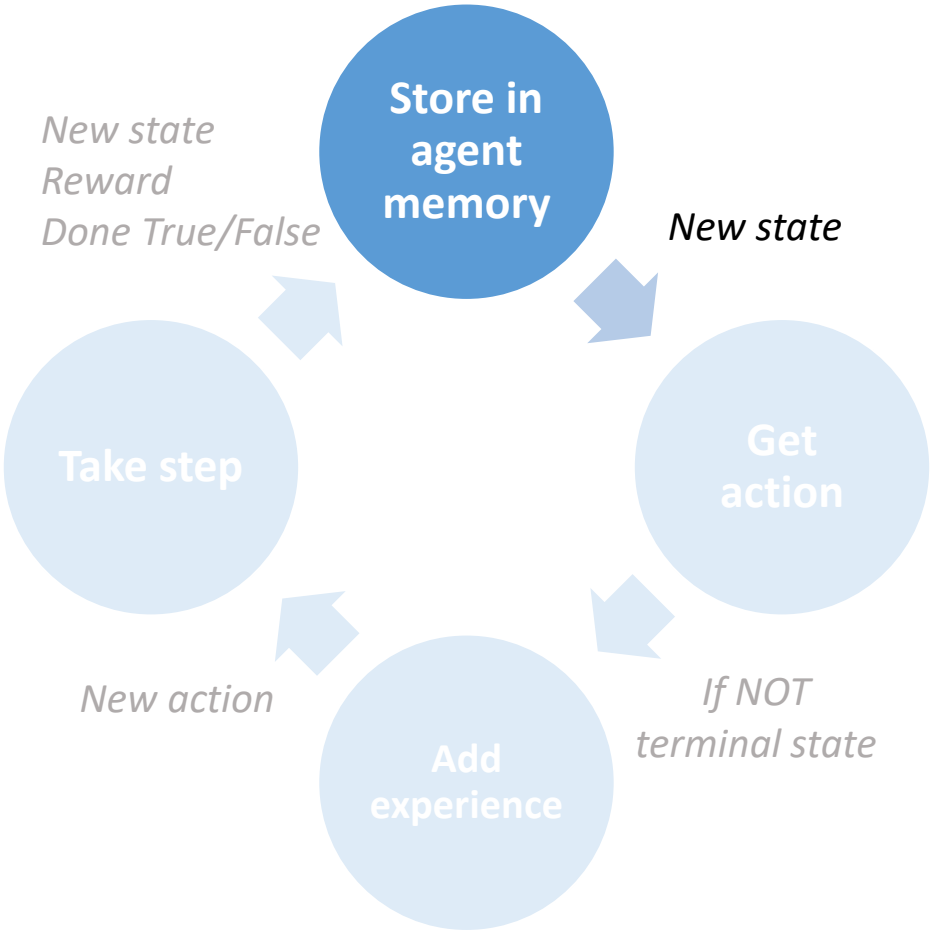


CS231N Project: Deep Q-Learning applied to air tactical deconfliction

Learning workflow

- ***Store in agent memory***: the implemented DQN agent stores the outputs from the .step() function, which will be employed for model updates.
- The new state is stacked with the previous three frames:



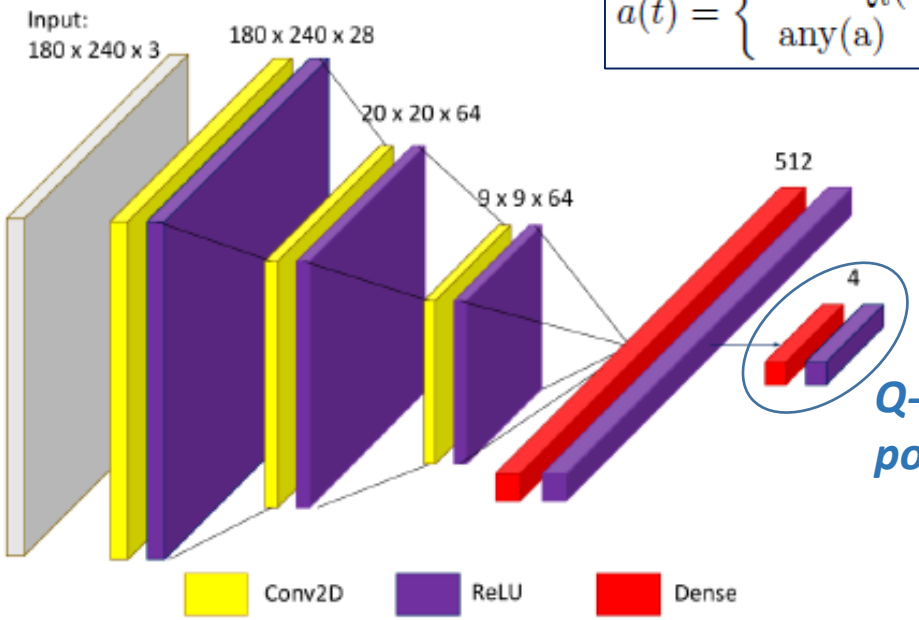


CS231N Project: Deep Q-Learning applied to air tactical deconfliction

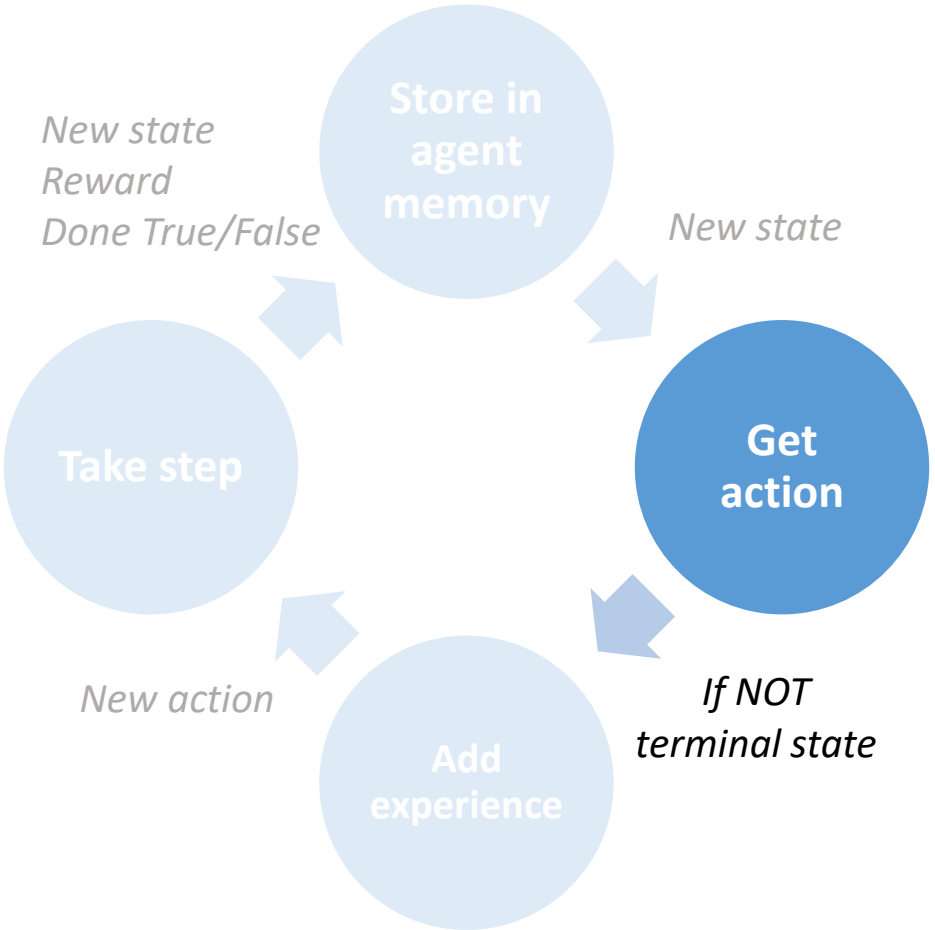
Learning workflow

- ***Get action***: given the new state, the CNN model is called in order to predict the next action.
- The action is chosen based on an ε-greedy policy.

$$a(t) = \begin{cases} \max Q_t(a) & \text{probability} \quad 1-\epsilon \\ \text{any}(a) & \text{probability} \quad \epsilon \end{cases}$$



Q-value for each posible action

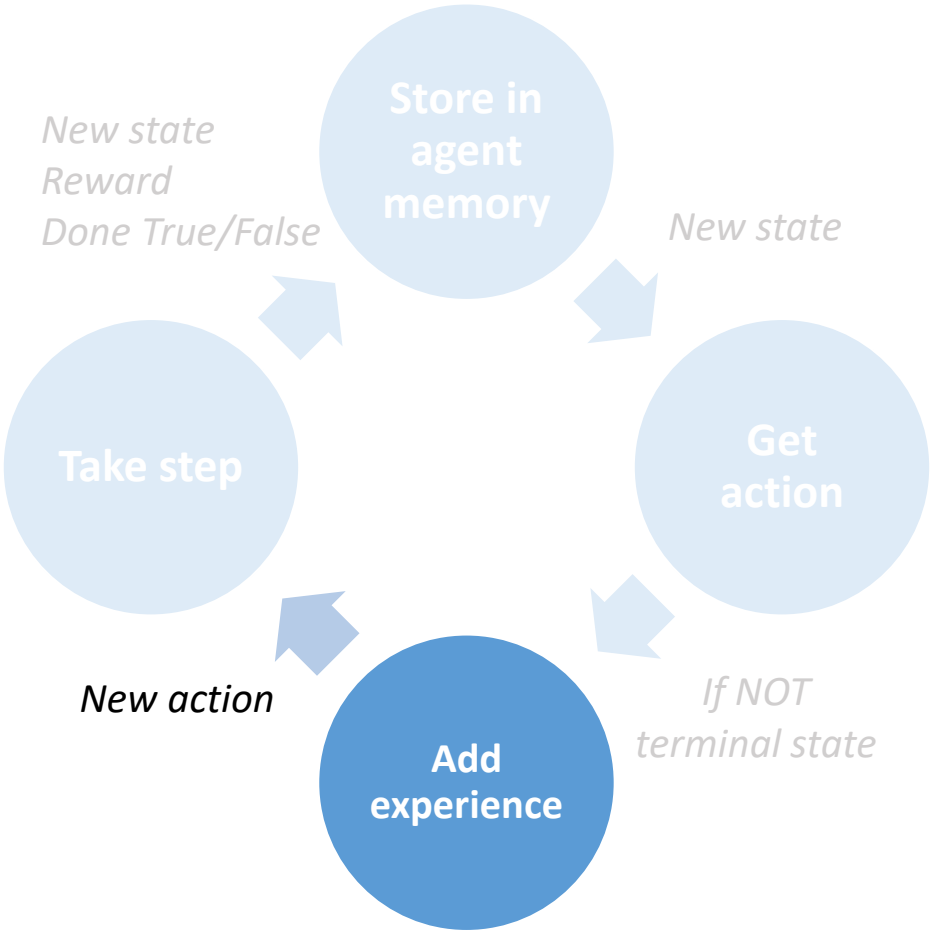


Learning workflow

- ***Add experience***: the model.fit() function is called in order to train the model with the experience stored from the last state.
- Summary of model hyperparameters:

Parameter	Description	Value
α	Learning rate	0.025
γ	Discount factor	0.95
ϵ_0	Initial ϵ	0.9
ϵ_{min}	Minimum ϵ	0.05
ϵ_{decay}	Decy value for ϵ	0.0009
β_1	Adam's 1st moment decay	0.9
β_2	Adam's 2nd moment decay	0.999

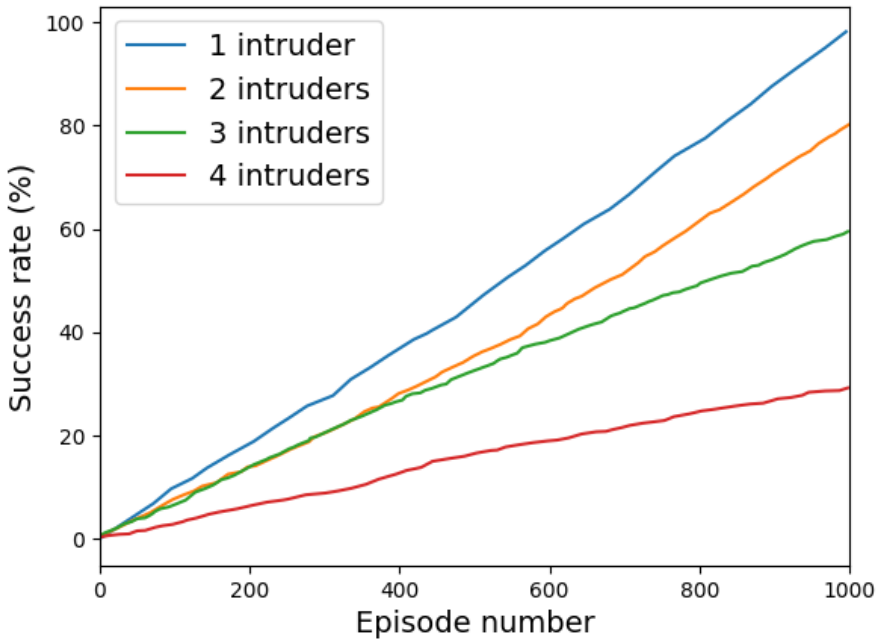
- The Q-values associated to each action and state after each step are updated.

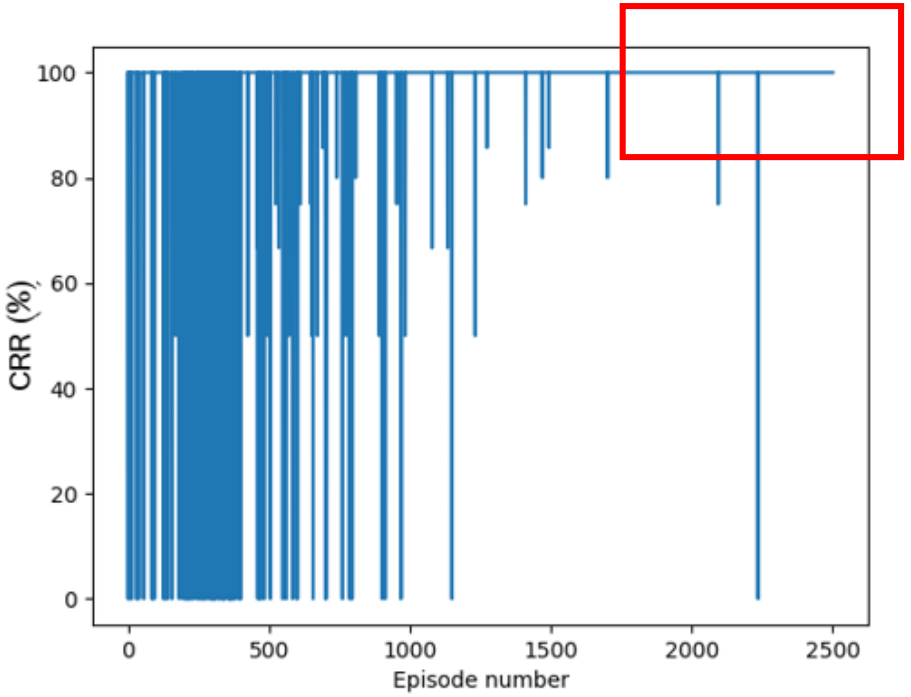


Evaluation: Relevant metrics

Two performance metrics were found to be relevant to the CDR problem:

- The Success Rate (%) is the percentage of simulations/episodes in which the ownship reaches the target.
- The Conflict Resolution Rate (%) is the percentage of conflict situations (separation distance lower than a threshold) that the model solves successfully.

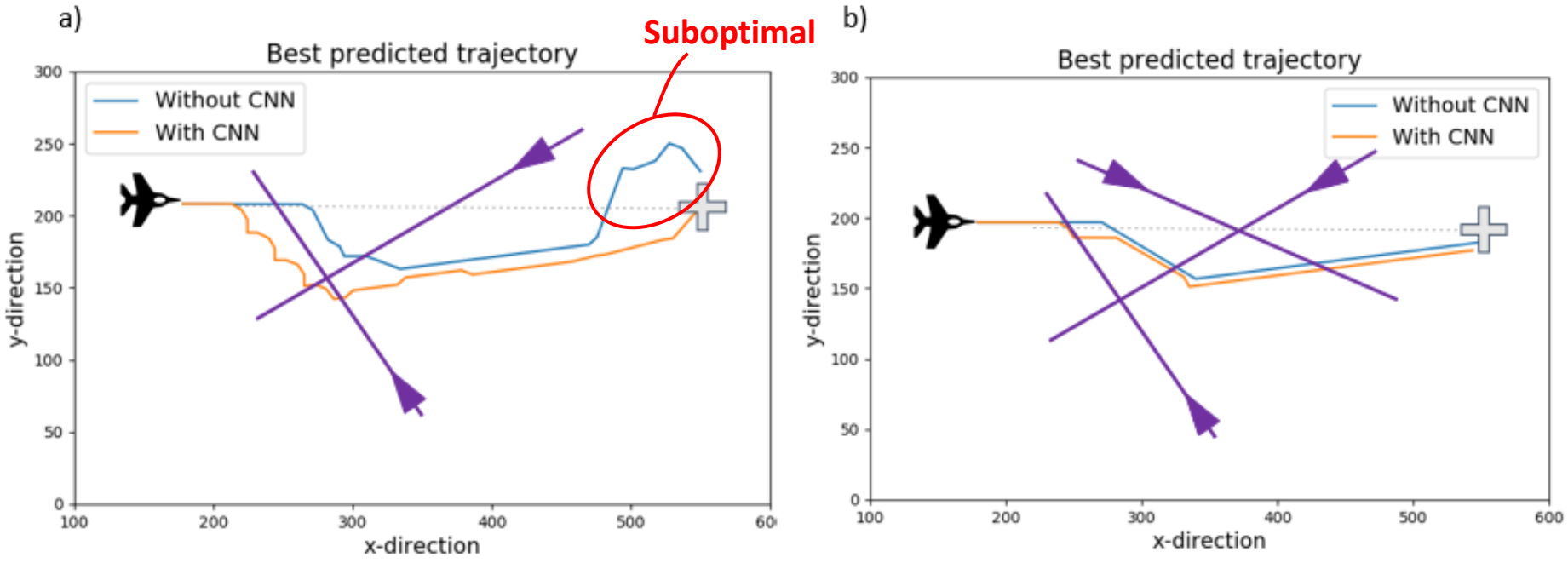




Important to get 100% CRR in order to enforce safe operations

Evaluation: Comparison with other methods (DQN without CNNs)

- Assess the benefits of using image representation + CNNs for this kind of problems → creation of an identical RL model using only state information as arrays. Removal of the convolutional layers.
- Similar performance on success rate (regardless the number of intruders the SR converges to 100% for both models).
- Evaluation of predicted best trajectories:



Deep Q-Learning applied to air tactical deconfliction

CS231N – Spring 2022

Marta Palomar Toledano
mpalomar@stanford.edu