

Reinforcement Learning Agent for Word Zapper (Atari)

Selective Topics in AI

Team Members

Mohamed Anwar Hosny	3	Abdelrahman Maged	2
Mohamed Ibrahim Abdelfatah	3	Abdallah Nasser Atta	3
Ahmed Raffat feshar	1	Fady Fayez	3
Thomas Basem	2		
Mohamed Ahmed Elnashar	3		

Project Idea

- *WordZapper* is a classic Atari game where an agent must shoot the correct letters to complete a word.



Introduction

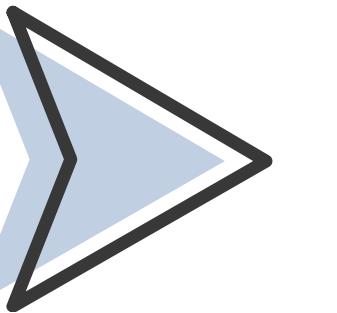
- **Word Zapper** is a classic Atari game where an agent must shoot the correct letters to complete a word.
- **Our goal:** Train an AI agent using Reinforcement Learning to play Word Zapper.
- **We implemented two approaches:** Deep Q-Network (DQN) and Proximal Policy Optimization (PPO).

Game Overview

- The agent controls a spaceship.
- Letters are at the top of the screen.
- The goal is to shoot the correct letter of a target word.
- Incorrect shots or wrong timing are penalized.



Available Actions



WordZapper has the action space Discrete(18) with the table at the right listing the meaning of each action's meanings

Index	Action	Description
0	NOOP	No operation, do nothing.
1	FIRE	Press the fire button without updating the joystick position
2	UP	Apply a Δ-movement upwards on the joystick
3	RIGHT	Apply a Δ-movement rightward on the joystick
4	LEFT	Apply a Δ-movement leftward on the joystick
5	DOWN	Apply a Δ-movement downward on the joystick
6	UPRIGHT	Execute UP and RIGHT
7	UPLEFT	Execute UP and LEFT
8	DOWNRIGHT	Execute DOWN and RIGHT
9	DOWNLEFT	Execute DOWN and LEFT
10	UPFIRE	Execute UP and FIRE
11	RIGHTFIRE	Execute RIGHT and FIRE
12	LEFTFIRE	Execute LEFT and FIRE
13	DNFIRE	Execute DOWN and FIRE
14	UPRIGHTFIRE	Execute UP and RIGHT and FIRE
15	UPLEFTFIRE	Execute UP and LEFT and FIRE
16	DOWNRIGHTFIRE	Execute DOWN and RIGHT and FIRE
17	DOWNLEFTFIRE	Execute DOWN and LEFT and FIRE
40	RESET1	Toggles the Atari 2600 reset switch, not used for resetting the environment

Main Action Space

The following regular actions are defined by the Action enum

Do Nothing

Move

Shoot

State Representation

Input:

Raw frames (images) from the environment.

Preprocessing

Resizing, grayscale conversion, and stacking.

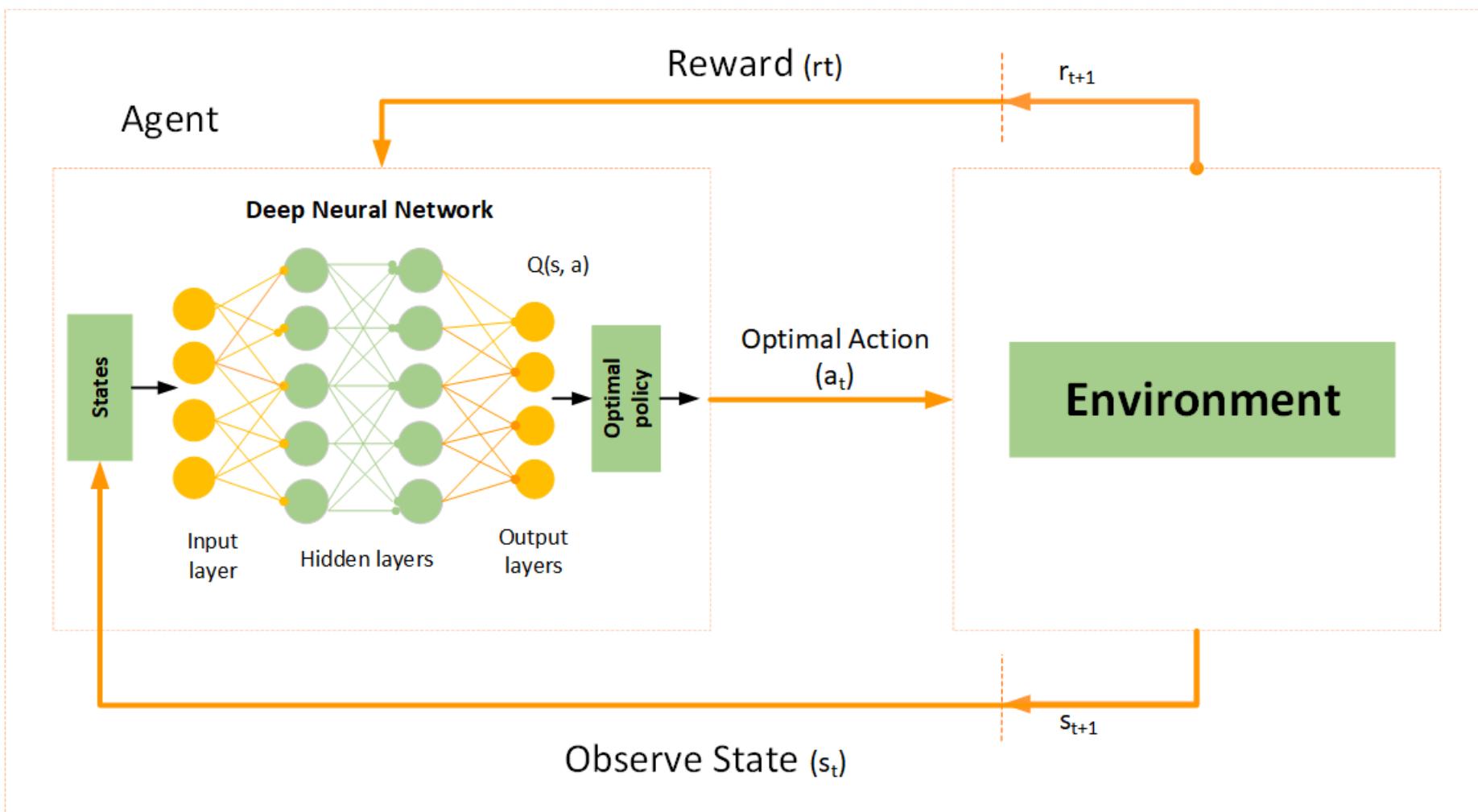
State:

Used as input to the deep learning model.

DQN Overview

- DQN uses a neural network to approximate the Q-value function.
- Implements experience replay and target network for stability.
- Epsilon-greedy strategy for exploration.

DQN Hyperparameters



Epsilon start:

1.0

Epsilon min:

0.05

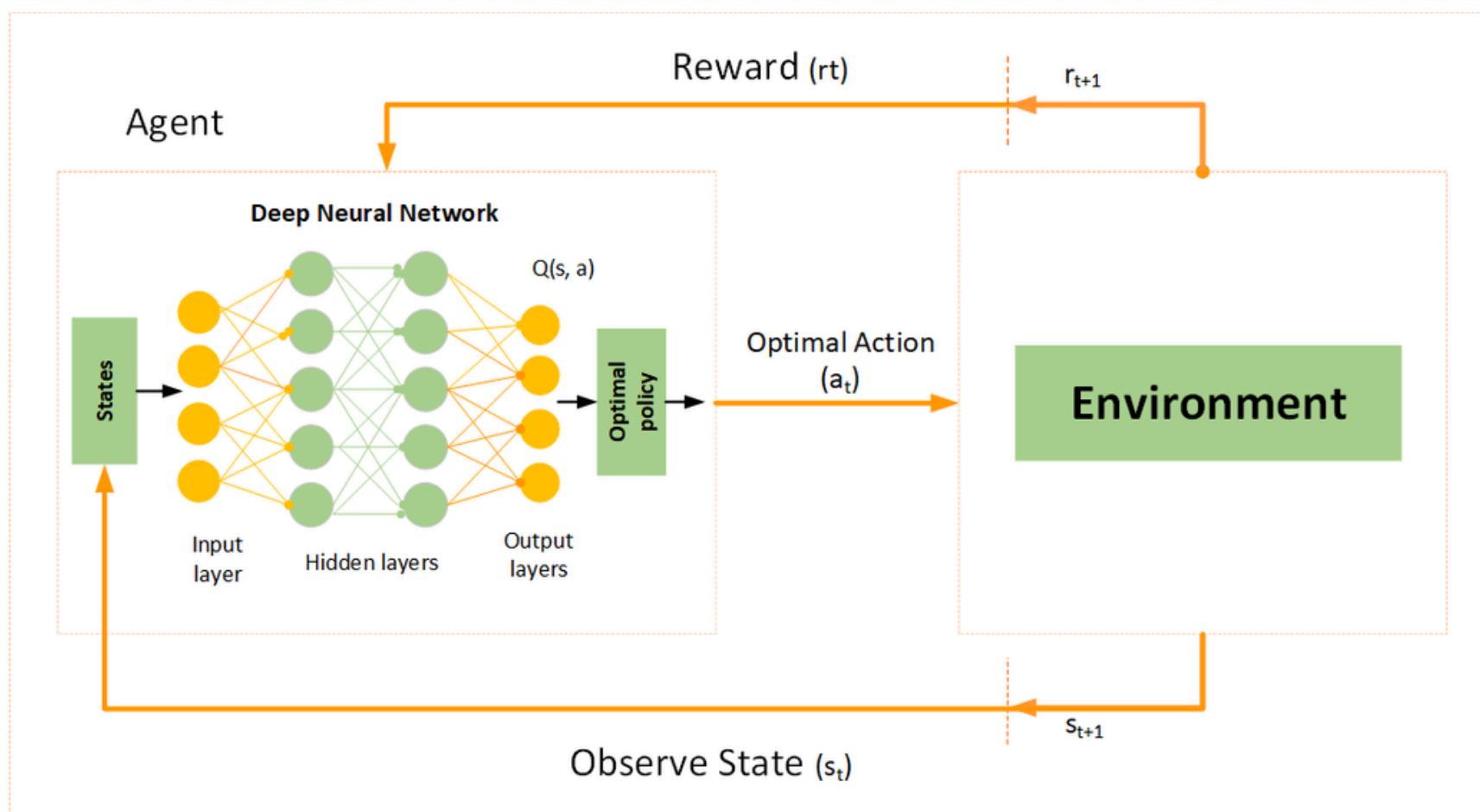
Epsilon decay:

0.995

Gamma (discount):

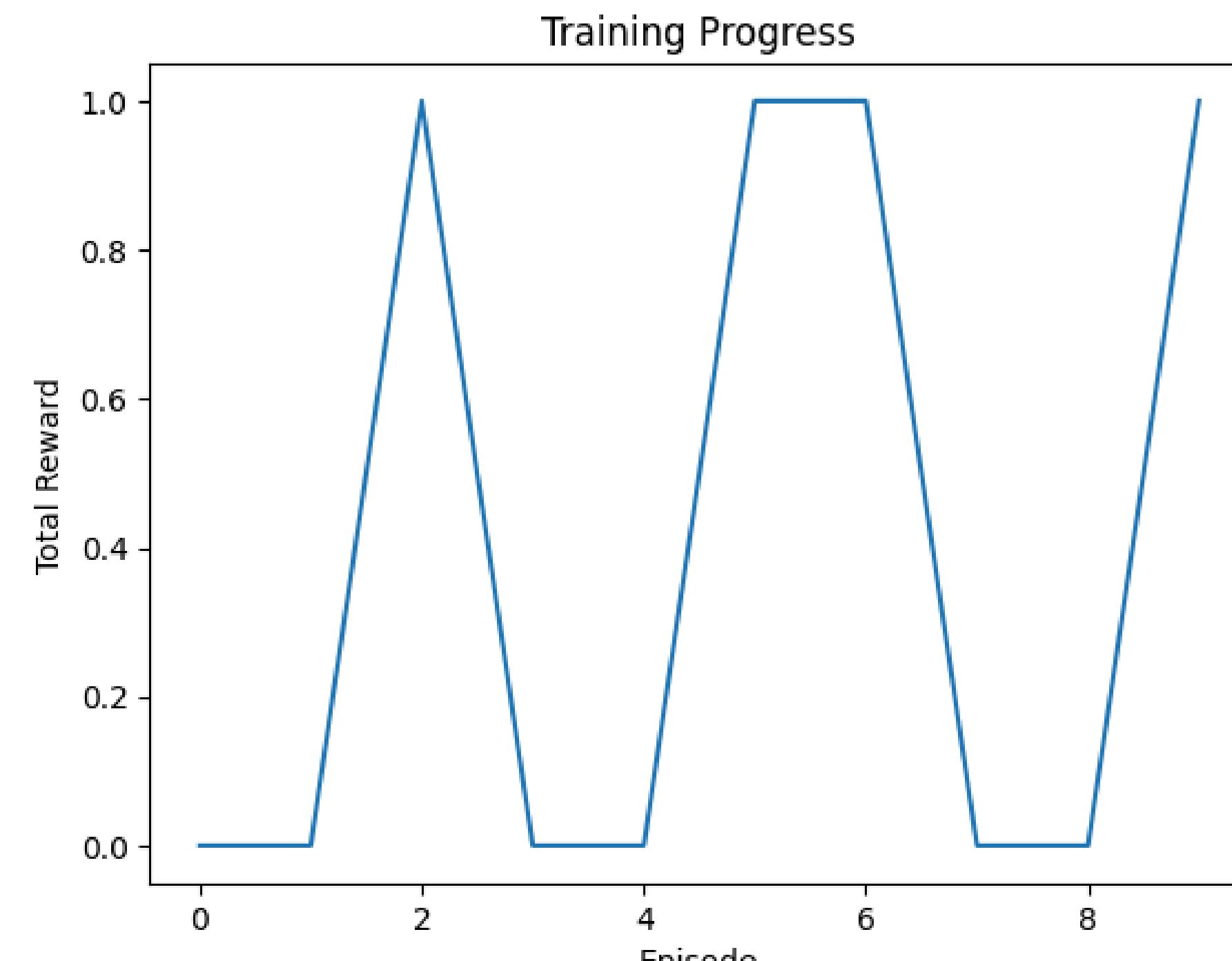
0.99

DQN Architecture



- ◆ Input: Processed image state
- ◆ CNN layers extract features
- ◆ Fully connected layers compute Q-values for each action

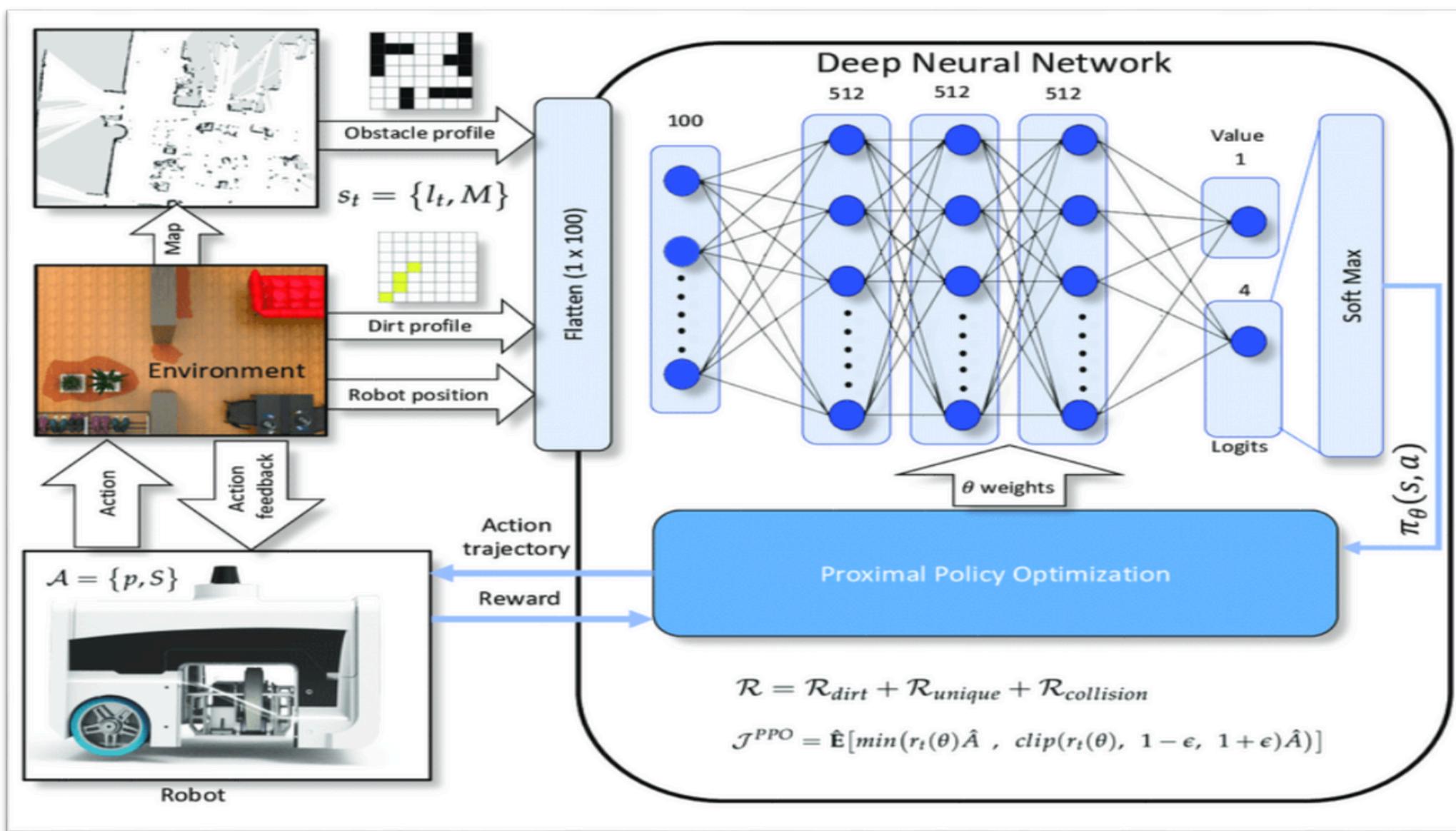
Visualization



PPO Overview

- PPO is a policy gradient method
- More stable and reliable than DQN
- Used CnnPolicy from Stable Baselines3 for image inputs

PPO Training Setup



Framework:

STABLE BASELINES3

Environment:

DUMMYVECENV WITH ATARIWRAPPER

Time Steps:

2,000,000

Policy:

CNN-BASED ACTOR-CRITIC MODEL

Model Comparison Summary

Brief Comparison between DQN and PPO

Feature	DQN	PPO
Algorithm Type	Value-Based	Policy-Based
Stability	Moderate	High
Learning Speed	Slower	Faster
Word Completion	Poor	Slightly Better

Challenges Faced:

- Complex visual input and dynamic word targets
- Sparse rewards delayed learning
- Letter extraction required image processing heuristics
- Frame stacking and preprocessing sensitive to performance

Summary

- Word Zapper is a challenging environment due to visual complexity and sparse rewards
- PPO outperformed DQN in stability and performance
- Agent learned low-level actions (shoot, avoid), but not higher-level goal (complete words)

Thank You!