SOC Final Project Report ‘25

Project Overview

My Final SOC Project aimed to train Agent Jackie, a Deep Q Learning (DQN) agent, to play the Atari game Kung Fu Master using OpenAI’s Gymnasium. The agent learned entirely from scratch, relying only on its interactions with the environment(no external data). It observed the game through raw pixel frames, chose actions like jumping or attacking, and gradually learned to survive and defeat enemies through reinforcement learning.

Approach Summary

First, raw game frames were preprocessed by converting them to grayscale and resizing them to 84×84 pixels to reduce input complexity.
To give the agent temporal context, the last four frames were stacked using *FrameStackObservation*, allowing the model to perceive motion and transitions.
The neural architecture was built using a Convolutional Neural Network (CNN) with three convolutional layers followed by dense layers, already given us in *helper.py*.
I employed an epsilon-greedy strategy for exploration, with a custom manual epsilon decay to fine-tune the balance between exploration and exploitation.
Experience replay was implemented with a buffer size of 100,000 to break correlations in the training data.
And,the target network was updated every 1000 steps to reduce the risk of divergence.
I also approached to save the best performance during training whenever a new highest reward was achieved.
Also integrated the video recording functionality made it easier to qualitatively verify improvements in gameplay and track the agent’s evolution over time.

Challenges & Fixes

1) Initial Setup Issues: Faced Gymnasium import problems, resolved by switching to a clean Miniconda environment.
2) ALE Not Detected: Installed `ale-py` to load Atari environments like Kung Fu Master but it’s not detected so i imported ale-py explicitly.

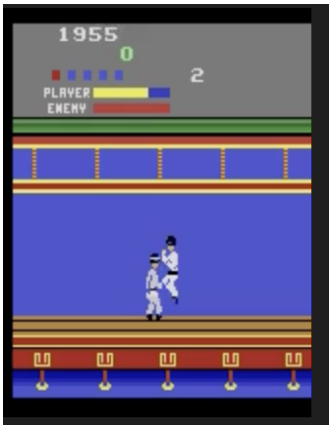
```
>>> import gymnasium
>>> env = gymnasium.make("ALE/KungFuMaster-v5", render_mode="human")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/opt/homebrew/Caskroom/miniconda/base/envs/agentjackie/lib/python3.10/site-packages/gymnasium/envs/registration.py", line 681, in make
    env_spec = _find_spec(id)
  File "/opt/homebrew/Caskroom/miniconda/base/envs/agentjackie/lib/python3.10/site-packages/gymnasium/envs/registration.py", line 526, in _find_sp
ec
    _check_version_exists(ns, name, version)
  File "/opt/homebrew/Caskroom/miniconda/base/envs/agentjackie/lib/python3.10/site-packages/gymnasium/envs/registration.py", line 392, in _check_v
ersion_exists
    _check_name_exists(ns, name)
  File "/opt/homebrew/Caskroom/miniconda/base/envs/agentjackie/lib/python3.10/site-packages/gymnasium/envs/registration.py", line 355, in _check_n
ame_exists
    _check_namespace_exists(ns)
  File "/opt/homebrew/Caskroom/miniconda/base/envs/agentjackie/lib/python3.10/site-packages/gymnasium/envs/registration.py", line 349, in _check_n
amespace_exists
    raise error.NamespaceNotFound(f"Namespace {ns} not found. {suggestion_msg}")
gymnasium.error.NamespaceNotFound: Namespace ALE not found. Have you installed the proper package for ALE?
>>> exit()
(agentjackie) lavanyasingh@LAVANYAs-MacBook-Air ~ % AutoROM --accept-license
AutoROM will download the Atari 2600 ROMs.
They will be installed to:
       /Users/lavanyasingh/.pyenv/versions/3.10.13/lib/python3.10/site-packages/AutoROM/roms
```

3) FrameStack Import Errors: Fixed incorrect import paths for `FrameStack` so fixed it after importing the FrameStackObservation because there is no FrameStack named wrapper exist.

```
(agentjackie) (3.10.13) lavanyasingh@LAVANYAs-MacBook-Air AgentJackieRL % python main.py
Traceback (most recent call last):
  File "/Users/lavanyasingh/AgentJackieRL/main.py", line 10, in <module>
    from gymnasium.wrappers import FrameStack
ImportError: cannot import name 'FrameStack' from 'gymnasium.wrappers' (/Users/lavanyasingh/.pyenv/versions/3.10.13/lib/python3.10/site-packages/gymna
sium/wrappers/__init__.py)
```

4) Epsilon Not Decaying: Manually implemented epsilon decay to ensure exploration is reduced over time.

```
Episode 4978 - Reward: 300.0 - Epsilon: 1.000
Episode 4979 - Reward: 600.0 - Epsilon: 1.000
Episode 4980 - Reward: 200.0 - Epsilon: 1.000
Episode 4981 - Reward: 500.0 - Epsilon: 1.000
Episode 4982 - Reward: 600.0 - Epsilon: 1.000
Episode 4983 - Reward: 600.0 - Epsilon: 1.000
Episode 4984 - Reward: 800.0 - Epsilon: 1.000
Episode 4985 - Reward: 1100.0 - Epsilon: 1.000
Episode 4986 - Reward: 200.0 - Epsilon: 1.000
Episode 4987 - Reward: 400.0 - Epsilon: 1.000
Episode 4988 - Reward: 600.0 - Epsilon: 1.000
Episode 4989 - Reward: 200.0 - Epsilon: 1.000
Episode 4990 - Reward: 500.0 - Epsilon: 1.000
Episode 4991 - Reward: 200.0 - Epsilon: 1.000
Episode 4992 - Reward: 500.0 - Epsilon: 1.000
Episode 4993 - Reward: 0.0 - Epsilon: 1.000
Episode 4994 - Reward: 100.0 - Epsilon: 1.000
Episode 4995 - Reward: 200.0 - Epsilon: 1.000
Episode 4996 - Reward: 0.0 - Epsilon: 1.000
Episode 4997 - Reward: 200.0 - Epsilon: 1.000
Episode 4998 - Reward: 800.0 - Epsilon: 1.000
Episode 4999 - Reward: 700.0 - Epsilon: 1.000
Episode 5000 - Reward: 200.0 - Epsilon: 1.000
```



Results

Despite the initial setbacks and issues,with all components working,Agent Jackie began learning effectively from experience. Agent Jackie demonstrated progressive learning across episodes. As training continued, the agent consistently improved its ability to survive longer and eliminate enemies more efficiently. Key milestones were captured using the *RecordVideo* wrapper, enabling a visual assessment of the agent's gameplay. The model checkpointing strategy ensured that the best-performing model was saved based on the highest reward observed during training, providing a reliable output for final evaluation.