

## DETAILED INSTRUCTIONS TO COMPILE AND EXECUTE THE DQN CODE

### *Dependencies*

The following Python libraries are required to run the DQN code for the Atari Breakout study. These dependencies ensure the environment, neural network, numerical operations, plotting, and image handling function correctly:

- `gym==0.21.0`: Provides the Atari Breakout environment for reinforcement learning.
- `torch==1.9.0`: PyTorch library for building and training the DQN neural network.
- `numpy==1.21.2`: Handles numerical operations and array manipulations.
- `matplotlib==3.4.3`: Used for plotting training progress, such as reward trends.
- `imageio==2.9.0`: Manages image input/output, useful for saving gameplay frames or videos.

Ensure your Python version is compatible (Python 3.63.9 is recommended for these library versions). Additionally, you need `pip` to install these packages and `git` to clone the repository.

### *Execution Steps*

- 1) **Clone the Repository**: Begin by cloning the repository containing the Deep Q-Network (DQN) code for the Atari Breakout study. Open a terminal and run the following command to download the repository to your local machine:

```
git clone https://github.com/your_dqn_repo
```

This assumes you have Git installed. If not, download and install Git from <https://git-scm.com/downloads>.

- 2) **Navigate to the Project Directory**: Change your working directory to the cloned repository's folder. Use the following command in the terminal:

```
cd your_dqn_repo
```

Ensure you are in the correct directory by listing the contents with `ls` (or `dir` on Windows) to verify that files like `train_dqn.py` are present.

- 3) **Set Up a Virtual Environment (Optional but Recommended)**: To avoid dependency conflicts, create a Python virtual environment. Run the following commands:

```
python -m venv dqn_env
source dqn_env/bin/activate    % On Windows, use: dqn_env\Scripts\activate
```

Once activated, your terminal prompt should change to indicate the virtual environment is active.

- 4) **Install Dependencies**: Install the required Python libraries listed above. Execute the following command in the terminal (within the virtual environment if you set one up):

```
pip install gym==0.21.0 torch==1.9.0 numpy==1.21.2 matplotlib==3.4.3 imageio==2.9.0
```

You may need to upgrade pip first with `pip install --upgrade pip` to ensure smooth installation.

- 5) **Run the Training Script**: Execute the training script to start the DQN training process on the Atari Breakout game. Use the following command:

```
python train_dqn.py
```

This script initializes the DQN agent, sets up the Breakout environment, and begins training. Ensure your system has sufficient memory (at least 8GB RAM recommended) and, if available, a GPU for faster training with PyTorch.

- 6) **Monitor Training Progress**: During execution, the script will log training metrics such as episode rewards and losses to the terminal. Additionally, the script generates and saves plots of the training progress (e.g., reward over episodes) to a file named `training_progress.png` in the project directory. Open this file with an image viewer to visualize the training trends. If the plot file is not generated, check the script for errors in the plotting section or ensure Matplotlib is correctly installed.

- 7) **Troubleshooting Tips**:

- If you encounter a `ModuleNotFoundError`, verify that all dependencies are installed in the active environment. Re-run the `pip install` command if needed.

- If the Breakout environment fails to load, ensure the Gym library is version 0.21.0, as newer versions may have compatibility issues with Atari environments.
  - For GPU-related errors, check that your PyTorch installation supports CUDA (if using a GPU). You can install a CPU-only version of PyTorch with `pip install torch==1.9.0+cpu` if needed.
- 8) **View Results:** After training completes, the script may save additional outputs like the trained model weights (e.g., `dqn_model.pth`) or gameplay videos, depending on the scripts configuration. Check the project directory for these files. To evaluate the trained model, you may need to run a separate evaluation script (e.g., `evaluate_dqn.py`) if provided, or modify `train_dqn.py` to include an evaluation phase.