AML_3406
AI and ML Capstone Project

FIGHTER AI MODEL USING REINFORCEMENT LEARNING

Group Name: Group-5

Group Members:

First Name	Last Name	Student Number
Abhinav	Duvvuru	C0810417
Akash	Panchani	C0813843
Sandeep Reddy	Alla	C0824160
Sivapriya	Tatireddy	C0817271
Uma Maheswara Rao	Gutta	C0814571

Submitted to: Professor Mr. Vahid Hadavi

Submission date: August 20, 2022.

Table of Contents

1. Abstract

In this project, we have set up a Fighter AI Model using the Reinforcement Learning. In this project, we have used the Gym and Gym-Retro packages in order to set up our fighter environment model. We have used ROM to set up the fighter model using the Gym-Retro environment. We have initially calculated the fight rewards to check how the fighter is performing based on the model. We have performed the preprocessing on the model to gray scaling, changing the pixels of the environment frame and reshaping the pixels of the model.

After performing the preprocessing steps of the model, we have started working on the hyper parameter tuning of the model in order to check the best model based on the rewards provided by the fighter performance. We are training the model that has the given the best performance based on the hyper parameter tuning output. We have setup call back functions in order to deal with the changing metrics of the fighter model. After training the model, we are evaluating the best model that we got from the tuning. We are taking the mean reward function as our metric for the model. We are testing our model based on the fighter performance.

2. Introduction

Playing games and sports is one of the most entertaining aspects of everyone’s life. This is what gives us the best opportunity to tune ourselves back to normal after working on various hectic things in our daily life. Playing and involving in different sporting activities will help us in relaxing the mind. Sporting activities will also help us in creating a better space for your body and mind. They will also help us in maintaining the body fitness by exercising and movement in the body.

We are now trying to learn a lot of new things from internet using mobile devices. It is all due to the increasing demand and drastic improvements in the field of technology as well as the reach of internet to all parts of the world. We are looking into the internet for all the day to day happenings across the world and improving our knowledge in various aspects without wasting more time and energy. As we have been using mobile phones and other technical devices for most of the time these days, we are also gaining lot of knowledge with minimal effort.

With the advent of commercialism and space issues in the metropolitan cities, it is increasing difficult and highly challenging to find places for kids to get proper facilities to play outdoor games and sports. The changes of them playing the traditional outdoor games like cricket and football are getting diminished due to this. Getting into paid coaching academies and facilities is not affordable by many middle-class families and kids. This is also one of the main reasons that kids and sports enthusiasts are turning their heads towards online games.

On one side, with the advent of technology and mobile phones, it became very easy for everyone to afford a mobile and get easy understanding of its usages. On the other hand, space issues became a major concern and acting as a major hurdle for playing outdoor sports. So, the online games are playing an important part in engaging the kids and other people to spend and invest their free time in the mobile and laptop games. This is easy for everyone to have a mobile and playing the games anywhere in journey and where ever you get 5 minutes of free time.

Due to increasing demand in the sporting sector for online games, there are lot of companies that are working on building various different sporting apps and online games. Like the competency with in companies to build best games, there is similar competency in users to play the games at the best of their ability. Reinforcement Learning helps us in building better sporting games according to your requirement. Our project is all about building the fighter AI model this is competent to the already existing street fighter models.

Our project is all about building a fighter AI model which will be competing with other models by tuning it necessarily. We are using the Gym and Gym Retro platforms of the reinforcement learning. We will be training the model according to our needs and constantly monitor the performance of the model after updating the training parameters. We are getting the model environment to be set up, then doing the preprocessing, followed by hyperparameter tuning and training the model to get the best model out, and finally we are evaluating the model to get the test results of the model fighter AI. The results can be better seen by training it for more time comparatively. After very iteration of the evaluation, we will get the better results followed by best results.

3. Models and Packages

We used various open-source python libraries like Gym, Gym Retro, Stable baselines, PPO algorithms, Optuna for the optimization and ROM for the set up of the environment. We are going to discuss more about them as follows:

3.1 Gym Library

We are using Gym for creating and setting up our model for the project. Gym is an open-source python library for handling and creating the reinforcement learning algorithms. It is providing the standard API for implementing the solution and setting up other environment specific needs. We are using the Env module for creating the fighter AI model in the project. The Gym is written in Python and is creating the platform for different models related to AI.

3.2 Gym Retro

Gym Retro is a platform specifically for the setup of games in the Python Gym environment. It is specifically designed to do the research on the games designed using reinforcement learning. We are using this environment to set up our environment with integration to Gym python library and in creating the basic model of fighter AI. There are more than 1000 games that are available in Gym Retro for integration. It is also used to train the classic video games initially and now also used to train them in Python.

We can get more idea about the Gym Retro platform by seeing the picture below that has the environment set up for the games.

Figure 1. Gym Retro Environment

3.3 ROM Insertion

ROM is very important in setting up the fighter model of our project. It helps us in integrating the fighter model with the game environment that we got from the Gym Retro. By using the ROM, we are combining the fighter along with the environment that is being used to fight. There are different types of ROMs available in the internet, but not everything is feasible for setting up the fighter environment. So, we have selected the ROM from the website https://wowroms.com/, which has different versions of the ROMs again, that are explicitly good for different regions across the world. We can see that from the below figure on where from where we got the ROMs from as follows:

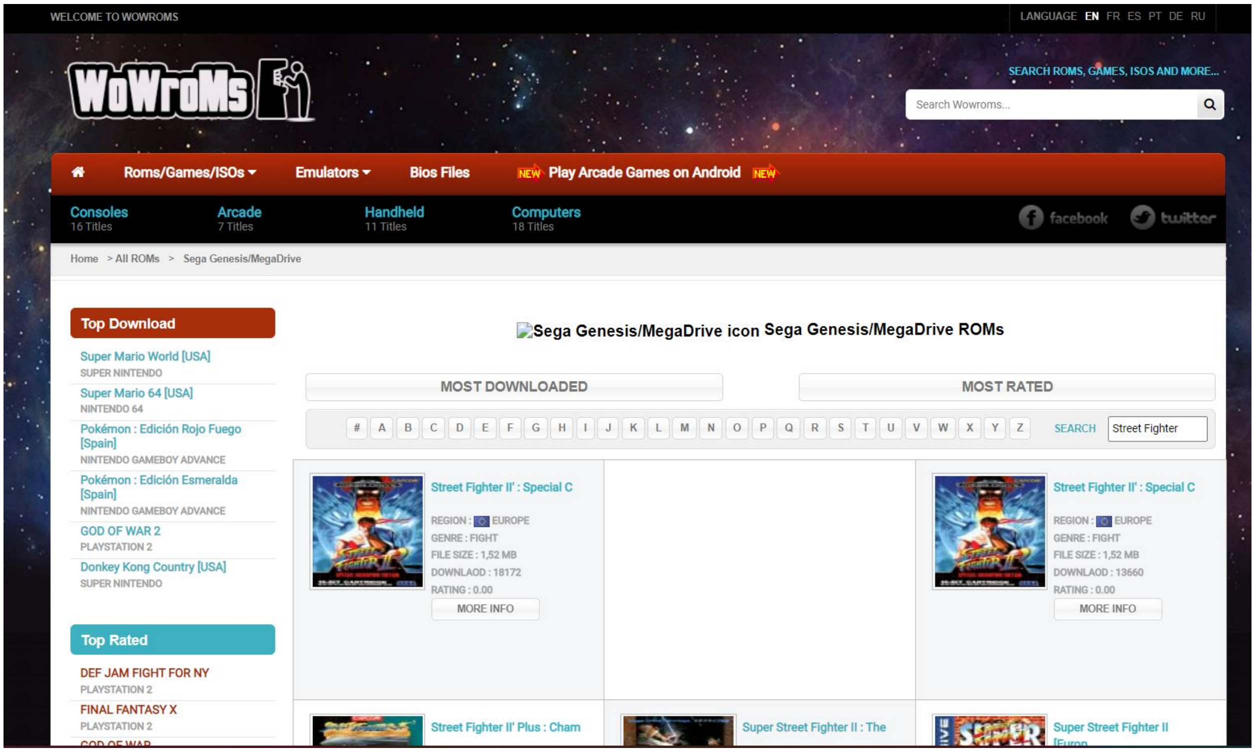

WELCOME TO WOWROMS
LANGUAGE EN FR ES PT DE RU
WoWroMs
SEARCH ROMS, GAMES, ISOS AND MORE...
Search Wowroms...
Roms/Games/ISOs
Emulators
Bios Files
NEW Play Arcade Games on Android NEW
Consoles 16 Titles
Arcade 7 Titles
Handheld 11 Titles
Computers 18 Titles
facebook
twitter
Home > All ROMs > Sega Genesis/MegaDrive
Top Download
Super Mario World [USA] SUPER NINTENDO
Super Mario 64 [USA] NINTENDO 64
Pokémon : Edición Rojo Fuego [Spain] NINTENDO GAMEBOY ADVANCE
Pokémon : Edición Esmeralda [Spain] NINTENDO GAMEBOY ADVANCE
GOD OF WAR 2 PLAYSTATION 2
Donkey Kong Country [USA] SUPER NINTENDO
Top Rated
DEF JAM FIGHT FOR NY PLAYSTATION 2
FINAL FANTASY X PLAYSTATION 2
GOD OF WAR
Sega Genesis/MegaDrive icon Sega Genesis/MegaDrive ROMs
MOST DOWNLOADED
MOST RATED
# A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
SEARCH
Street Fighter
Street Fighter II' : Special C
REGION : EUROPE
GENRE : FIGHT
FILE SIZE : 1,52 MB
DOWNLAOD : 18172
RATING : 0.00
MORE INFO
Street Fighter II' : Special C
REGION : EUROPE
GENRE : FIGHT
FILE SIZE : 1,52 MB
DOWNLAOD : 13660
RATING : 0.00
MORE INFO
Street Fighter II' Plus : Cham
Super Street Fighter II : The
Super Street Fighter II [Europ

Figure 2. ROM setup

3.4 Stable Baselines3

Stable Baselines is used for the implementations of the algorithms for reinforcement learning along with the installations from Open AI environments. This is kind of like sklearn for the Machine Learning algorithms. It consists of different algorithms that are necessary for running the reinforcement learning problems. Stable Baselines3 is the latest and updated version of the current Stable Baselines package. We are using this in our project to deal with the creation of our fighter AI model. From this Stable Baselines3 package, we are using the PPO algorithm for training the model.

3.5 Proximal Policy Optimization (PPO)

Proximal policy optimization which is popularly known as PPO is a Reinforcement Learning algorithm. It is a policy gradient method which is used in the reinforcement learning models. It is used for the data efficiency and stable performance of the model. It is used for the first order optimization of the models with greater benefits. We are using the PPO algorithm from the Stable Baselines3 package for training our fighter model. This is because it takes very less time in giving efficient results for the model.

3.6 Optuna Framework

We are using the Optuna Framework for the optimization of our model. It is specifically used for the automatic optimization of the hyper parameter tuning process in reinforcement learning. It uses different parameters for optimizing using different approaches like Grid search, random search and Bayesian Search. We are using this for the process of hyper parameter tuning in our fighter AI model. It uses Tree structured parameters for the tuning of the model and also uses Gaussian as well for the process.

4. Fighter AI Model

Now, we are into the setup of our fighter AI model. We are going to perform the environmental setup followed by the preprocessing, hyper parameter tuning, training the model, evaluating the results and testing of the model.

4.1 Fighter Environment Setup

We are setting up the fighter AI model using the packages Gym and Gym Retro. Gym acts as a platform to create the reinforcement learning model. Using Gym Retro, we are taking the Street Fighter game environment from all the available 1000 plus games of Gym Retro. We are creating an environment and we need to set up the fighter for the environment with the help of ROM. We are using the United States version of the ROM from the Wow ROMs known as **'StreetFighterIISpecialChampionEdition-Genesis'**. With this we are creating the fighter along with the environment along with the fighter to work on the basic model of fighter AI. The below figure shows how our model is created with the environmental setup and fighter setup.

Figure 3. Fighter AI Model and Environment Setup

4.2 Model Preprocessing

Once the environment is setup completely, we have a basic model that is good at fighting the opponents but not as we have expected. It was still a raw model and we need to improve the model by doing some preprocessing and the parameter tuning of the model. So, we are doing the preprocessing tasks of the data. In the model preprocessing, we are going to perform the tasks like gray scaling the model data, reshaping it to a better parameter, resetting the model to initial states once the game is completed, modifying the reward function and setting up the frames in the data. The above tasks are done to preprocess the model into better outfit.

Coming the gray scaling of data, we are converting the model frames that are in BGR format to RGB format for conversions. After conversion, we are resizing the model to the size of 84*84 and adjusting the same as the channel size in the model. We are also resetting the model to initial when ever we are closing the environment. We are taking the previous frame as the observation and again taking the game forward from the next frame by setting the score to 0. We are now creating a step for the game by performing the action and previous frame is considered as the observation. We are also reshaping the reward function by the difference of the previous scores and new scores. We are creating a Class in python and taking these steps as different attributes and methods of the Class.

For every game, we get the observations, information, rewards and the done status of the model. The fighter will give us this information in every aspect of the game and we need to store them as preprocessed values and use them accordingly whenever they are required. The information space provides the observation of the fighter, that is the state of the fighter currently. Rewards will be provided based on the actions of the user. Information is provided regarding the model from the info attribute and the status of the action is stored by the done attribute of the model.

4.3 Hyper Parameter Tuning and Optimization of the Model

From the above preprocessing of the model, we are now trying to create the model based on the best parameters of the model. In order to find the best parameters for the model and to train the model, we are creating a function with different set of parameters required for the model. We are varying the basic parameters like steps of the model, gamma parameter of the model, the learning rate at which our model operates, clip range of the model as well the parameter lambda. All these parameters are varied from different levels of numerical values, so that we can achieve the best parameters out of them. We are trying to keep the learning rate as slow in order to meet the best standards of the model and also avoiding the fact of overfitting the model and training the parameters.

We are now tuning creating a function to set up the model for training. We are using the Proximal Policy Optimization (PPO) model of the Stable Baselines3 package of the reinforcement learning. This is one of the best algorithms of the reinforcement learning problems. We are setting up the model using PPO with different number of timesteps to check on the performance of the model. We are using the Pytorch to create this model and operating it based on the pytorch framework.

We are creating the model with very less learning rate as we need more detailed training for our model. We are not willing to converge the model very fast but even with the slow learning rate, the PPO model will create best and fast results. Now, we are using the Optuna framework to integrate the model with hyper parameter tuning of best parameters. This helps us in finding the optimized models and we can get the best model out of the existing models from this optimization. We are taking 10 trails for our training models and out of 10, we are selecting the best model based on the mean reward that the models achieved.

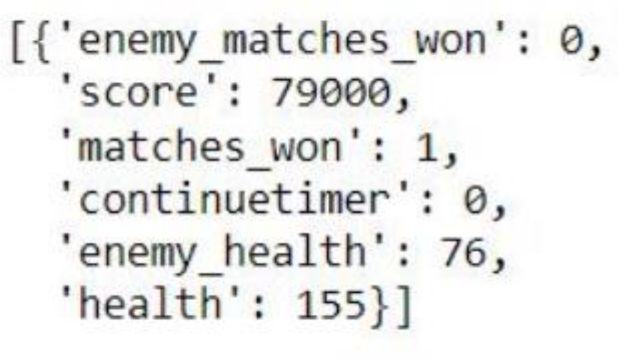
4.4. Training the Model

From the hyper parameter tuning and optimization, we got the best model out of all the 10 available models in the tuning process. The model with highest mean rewards is selected as the best model for further training. Now we are training the best model with further best training parameters. We are using the model created based on the Proximal policy Optimization algorithm of the reinforcement learning. We are trying with the greater number of steps this time and also varying the learning rate accordingly. We can use total timesteps on a large scale to develop the best model but there are space and other constraints that will come up with huge number of timesteps for the model. This is kind of learning the agent of the model and training him for better fights and opponents. We are also using the os module to save the model and import it when ever we need to run the best model at that point of time. So, from the training of the model, we will get some results and we will do the evaluation and testing for checking the results of the model.

4.5. Evaluation and Model Testing

Now that we have created and trained our model, we are going to check and evaluate the model based on the performance of the fighter. We are using the Proximal Policy Optimization algorithm's evaluate policy to evaluate the model. The mean reward function is our parameter and the model with highest mean reward will be like the best output of the model. We will then test the model based on the results of the evaluation that if the model is running fine or not.

We have got the mean reward of the model as "6700" and compared to initial basic model, we have got the best results for the model. One of the best results of the training model is looking as follows.



```
[{'enemy_matches_won': 0,
  'score': 79000,
  'matches_won': 1,
  'continuetimer': 0,
  'enemy_health': 76,
  'health': 155}]
```

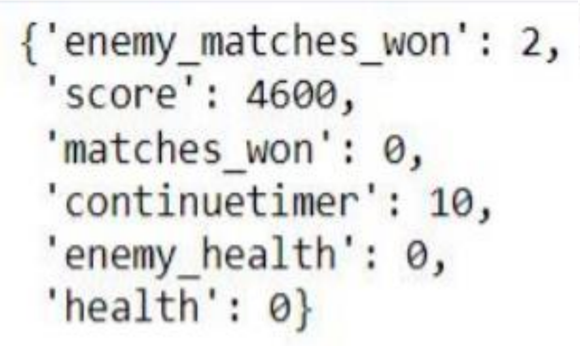
Figure 4. Fighter AI Model after training

The testing model evaluations are shown as above. The score of the model, the enemy won matches, matches won by us, the health activities of the enemy and our model and also the episode details of the model.

5. Results

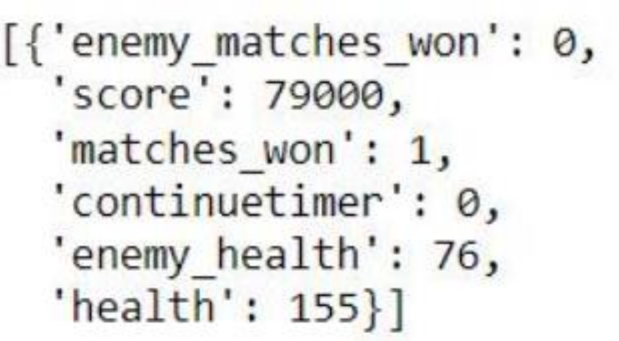
Finally, our model is trained and giving some of the best results in comparison to the model we have initially. The mean reward function of the model is given as the 6700 for the trained model of our project and some of the best results were showed in the above testing and evaluation slot of the project. It is the observation that we have done in our model that we performed the best in training the fighter.

From the below figures 5 and 6. We can certainly say that our model performed pretty good with better results



```
{'enemy_matches_won': 2,
 'score': 4600,
 'matches_won': 0,
 'continuetimer': 10,
 'enemy_health': 0,
 'health': 0}
```

Figure 5. Fighter AI Model before training



```
[{'enemy_matches_won': 0,
  'score': 79000,
  'matches_won': 1,
  'continuetimer': 0,
  'enemy_health': 76,
  'health': 155}]
```

Figure 6. Fighter AI Model after training

6. Conclusions and Future Work

In conclusion, we can say that the fighter model we have trained using the reinforcement learning through our project is working extremely well and can be used to fight the opponents quite easily in comparison to the initial models. We can also investigate the mean rewards of the fighter model in order to make better decision making using the optimization techniques.

Coming to the future work, we are planning to deploy the model using the Amazon Web Services and present it in a better way for users. We are also planning to implement the User Interface and deploying the solution as part of future work.

7. References

Gym Retro. (May 25, 2018). Retrieved from https://openai.com/blog/gym-retro/

Street Fighter II. (n.d.). Retrieved from https://wowroms.com/en/roms/list/Sega+Genesis%2FMegaDrive?search=Street%20Fighter

Stable Baselines docs (n.d.). Retrieved from https://stable-baselines.readthedocs.io/en/master/

Getting started with OpenAI Gym: The Basic Building Blocks (n.d.). Retrieved from https://blog.paperspace.com/getting-started-with-openai-gym/

StreetFighter II is hard, so I trained an AI to beat it for me (May 20, 2020). Retrieved from https://towardsdatascience.com/street-fighter-ii-is-hard-so-i-trained-an-ai-to-beat-it-for-me-891dd5fc05be

How to teach AI to play games: Deep Reinforcement Learning (Nov 15, 2018). Retrieved from https://towardsdatascience.com/how-to-teach-an-ai-to-play-games-deep-reinforcement-learning-28f9b920440a