Computer Games Development SE607 Software Functional Specification (Draft) Year IV

Adrien Dudon
C00278154

[Date of Submission]

[Declaration form to be attached]

Contents

1. Acknowledgements

Section to be completed.

2. Introduction

2.1. Intended audience and Reading suggestions

This document is addressed to every person willing to understand in more depth the software part of this research project.

The software described in this document is in support of the research project intituled: *What are the benefits of using Vision Transformers over Convolution Neural Networks for modern computer games?* The goal of this project is to compare three artificial neural networks against each other on video game environments, and to discover the effectiveness and benefits of one against another to solve a specific problem. For more details on this project, please look at the project report[99].

2.2. Purposes

The software for this project consists of multiple software and scripts. The goal is to compare three different model of neural networks against each other on different game environment. The compared neural networks are as follow: **traditional Artificial Neural Networks**, **Convolutional Neural Networks**, and **Vision Transformers.**

The purposes of the software programs developed for this project are to support the neural networks comparison and assumption made and explained in the research paper. It should be able to give different information regarding a neural network performance, its benefits and its effectiveness compared to other.

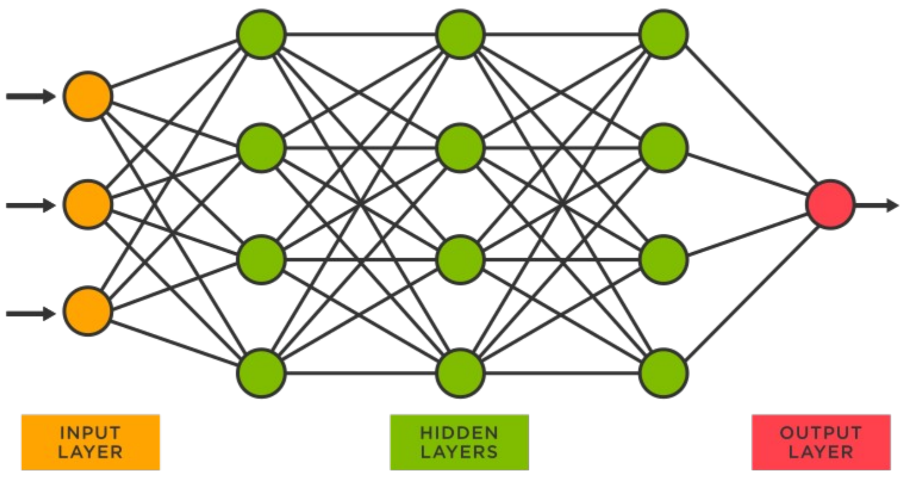

Figure 1: neural networks illustration

3. Functional Specification

The software goal is to train agents using reinforcement learning inside a game environment. The reinforcement learning model used is a Q-learning model and its variations, deep Q-learning and double Q-learning.

3.1. Introduction

This section describes the general introduction and workflow usage of the software without being too specific into features.

3.1.1.Reinforcement learning model

As explained before, the user of the software should be able to train an agent with reinforcement learning. And the reinforcement learning model choose for this task, is the Q-learning model.

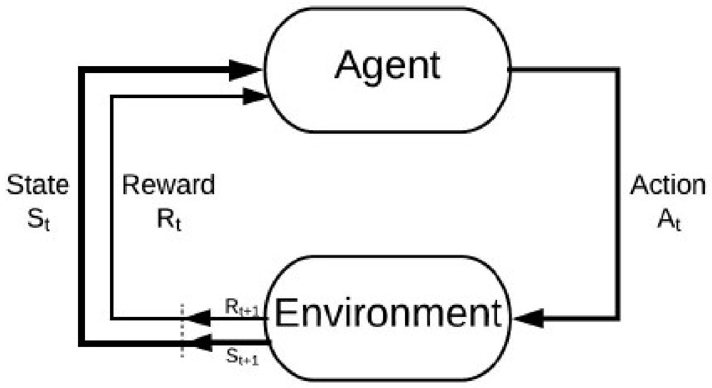

Figure 2: representation of the Q-learning model

3.1.2.Neural networks

The reinforcement learning problem is solved by a neural network. In the context of this project, three neural networks must be used to train the agent:

- Traditional Artificial Neural Network,
- Convolutional Neural Network (CNN),
- Vision Transformer (ViT),
- CNN and ViT combination.

The user should also be able to train the agent by combining CNN and ViT.

The software must allow the user to choose which neural network they would like to use to train the agent.

3.1.3.Environment

The software lets the user choose on what environment is the agent going to be trained on (e.g., Cart Pole). Simple and more complex environments could be used. But the software should at least feature the Cart Pole environment.

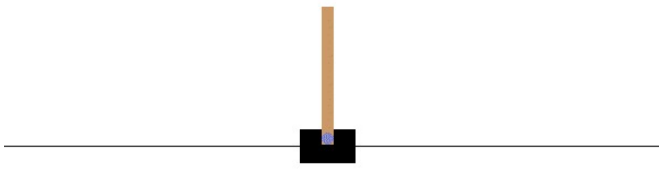

Figure 3: Cart Pole Gymnasium environnement

If the software feature more than one environment, the user should be able to choose from a list of environments. Having multiple environments is not a hard requirement for now (December 2022).

3.1.4. Hyperparameters

Before training, the user must be allowed to choose the hyperparameters to be used for the learning phase. Hyperparameters varies depending on the neural network used for training.

3.1.5. Training

The user should have control over the training process, it can either start, pause, or cancel it. The paused trained neural network can also be saved somewhere as a file and can be imported back into the software to continue the training later.

3.1.6.Training output

Training should generate useful data for comparison purposes. The output for the training is a video, followed by some useful tables and graphics, showing how the training performed over time. The trained agent must also be saved and stored somewhere, so it can be used later by the software.

The output data must be stored in a file and can be viewed by the software. Specific TensorFlow data can be viewed using TensorBoard (https://www.tensorflow.org/tensorboard).

3.1.7.Neural networks testing

Trained agent can be tested on the environment to see how they perform. It is basically a view on the environment, where we can see the agent (AI) playing the game or solving the environment.

3.1.8. Data comparison

The main purpose of this project is to compare neural networks against each other. As such, the software must allow cross data comparison between trained neural networks. Comparison is limited to its environment scope (it would not make sense to compare two unrelated trained agents on two different environments).

3.2. Gymnasium

The software will mainly use the environment library of Gymnasium and its API. Gymnasium features standard environment and standard API for reinforcement learning that facilitates the research on RL subjects using standard Python game environments. Each of the Gymnasium environment are RL problems that must be solved.

3.3. Functionality

This section presents an exhaustive list of all the features required by the final software.

Section to be completed.

3.3.1. Game environment

- The software should feature the Cart Pole environment from Gymnasium

3.3.2. Training

- On program start, user should be prompted to train the agent
- The training shall be paused by the user
- Training shall be cancelled by the user
- Training parameters shall be modified by the user
- The user can choose a name for its trained agent
- Multiple trained agents can exist for one game environment

4. References

Section to be completed. 99 are used as placeholder, because this section will probably be updated later on.

Report

[99] Dudon, A. (2022). *What are the benefits of using Vision Transformers over Convolution Neural Networks for modern computer games?* Carlow: South East Technological University.

Website

[99] Farama Foundation. Gymnasium. [Online]. (https://gymnasium.farama.org/). (Accessed 13 December 2022).

[99] Farama Foundation. The Farama Foundation. [Online]. (https://farama.org/). (Accessed 13 December 2022).