

Initial Plan: VizDoom AI via Deep Reinforcement Learning

CARDIFF
UNIVERSITY

PRIFYSGOL
CAERDYDD

CM3203, One Semester Individual Project

40 Credits

School of Computer Science and Informatics

Author: Aaron James
Supervisor: Frank C Langbein
Moderator: Steven Stockaert

Project Description:

In this project I will attempt to create an artificial intelligence agent that can play Doom via a Q-learning or Actor Critic approaches. I aim to produce an agent that can play any environment provided by the VizDoom module, potentially even custom maps, to a human or above human level, or may only be able to operate on few or a single map provided by VizDoom.

I have chosen to do this project in Python because I feel I understand the language better than others which, since I'm tackling something I'm already out of my comfort zone with, would give me a head start on understanding syntax. It also has good modules to support my project that I have researched or previously used throughout my degree. These include but are not limited to:

- VizDoom for environments, controls, frame buffer, among other functionalities.
- TensorFlow to produce a neural network to decide what action the AI should take, given a convolutional neural network is available in the module.
- Matplotlib to produce graphs for analysis and evaluation of the final algorithm.
- Numpy to provide additional tools for analysis.

I aim to produce a large body of source code detailing my deliverables. I would prefer to do it in source code so that it can be studied through on the Cardiff University website should I choose to publish it by myself, colleagues, or anyone working on making their own agent elsewhere.

VizDoom is built upon an open-source remake of the Doom game which provides many lower-level functionalities. I can avoid generating my own Doom environment via VizDoom scenarios and functions which should speed my project along. such as getting a frame buffer, which is a full pixel by pixel image of the current frame displayed on screen.

These functionalities include getting a frame buffer, which is a full pixel by pixel image of the current frame displayed on screen. The frame buffer can be used by a convolutional neural network to discover what is on screen and decide a course of action to yield the best result for that frame. With an agent that explores, such as a minimax-esque function, the agent could optimise for future frames too. The controls are also managed by VizDoom making it a very neat and straightforward package. On top of that, setting up VizDoom seems simple.

Using some base environment and random button presses can be done with little effort. The main challenge of this project is to expand upon this random button pressing into a much more competent agent with interesting architecture. I will attempt an Actor-Critic approach since I already have an environment with controls. I should make use of the environment which a Q-learning approach does not. Using Q-learning I could attempt to optimise various convolutional neural networks by adjusting number of layers and different types of node layers to increase the rate of learning on top of time taken to complete a level or score on that level.

I think this project would be more interesting if the algorithm optimised for time which would allow me to compare my AI's results to Speedrun.com's times or my own should my program not compare well to some of the best players. I could plot the timings on levels or milestones on a graph via Matplotlib along with rate of learning, final score per level vs the average among other analytics. Optimising for time or score may not be feasible however since Actor-Critic approaches learn both a policy function and a value function simultaneously.

I could create a database of times of runs of the doom WAD I choose to use. I purchased The Ultimate Doom on steam to gain access to its WAD which is the level file. By storing information about player's runs from speedrun.com anonymously or my own, I could compare it to the agent's learning, progress, and best runs. I could even track an individual's performance and see how a human player compares to the agent. I would need to verify if I could do this under GDPR and whether this information is reasonable and ethical to store through school policy and the ethics board.

Project Aims and Objectives:

In this project I aim to produce:

- Source code to run VizDoom as a project and create an agent to play it.
- Analytical graphs and stats about my program that shows how well the system learns, what it has learned to optimise, and how well it does to player models.
- A database of information about how human players and my algorithm played Doom for comparison as an extension.
- A 25000-word report detailing this project in its entirety.

Work Plan:

On a weekly basis, I intend on completing the following goals:

on a weekly basis, I will end on completing the following goals:

		Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Initial Report:	Initial Report						
Final Report:	Introduction						
	Background						
	Approach						
Implementation:	Get VizDoom + random button pressing working						
	Convolutional neural network with TensorFlow						
	Policy + value function definitions						
	Runtime for neural network to learn						
Results and Evaluation	Write analytics code with Numpy and Matplotlib						
	Discuss findings in report						
	Future Work						
	Conclusions						
	Reflection on learning						
	Additional						

		Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Key:
Initial Report:	Initial Report							Completed
Final Report:	Introduction							In progress
	Background							Not started
	Approach							
Implementation:	Get VizDoom + random button pressing working							
	Convolutional neural network with TensorFlow							
	Policy + value function definitions							
	Runtime for neural network to learn							
Results and Evaluation	Write analytics code with Numpy and Matplotlib							
	Discuss findings in report							
	Future Work							
	Conclusions							
	Reflection on learning							
	Additional							

The week where a task transitions from yellow to green is a week where that task gets completed. Either by or on this week. Overall I think this plan lays out all my tasks in a easy to understand workflow and shows clearly what I need to do when with plenty of time at the end as shown in image 2 of my Gantt chart. Weeks 10 11 and 12 are all empty which gives me space to move these around. I expect to hit several roadblocks while coding since I haven't used TensorFlow or VizDoom before. There will definitely be intricacies I don't understand and will take time to work out, therefor having space to move around 3 weeks is ideal. That said, I believe the plan to be mostly feasible in the given timeframe without anything tripping me up.

On a weekly basis, my schedule will be as follows:

Weekly approach:	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday		Key:
Work									Active
Meeting									Not active
Rest									

I intend on taking Wednesday and Sunday off work to rest. I intend on keeping Thursdays for work and meetings since there will be 4 days to make use of the advice given to me by my supervisor. I believe this will be an effective method of avoiding burnout whilst staying productive with an ideal timing for the meetings. I will likely use Thursday as a meeting day and use the day to work on another module, but depending on how my workload was managed for the week I will allocate additional days to my other module where my goals have been completed.

For deliverables, I will produce a program that can play Doom with a convolutional neural network via an Actor-Critic approach. I will produce various interesting statistics graphs that may or may not update in real time while the algorithm runs. Finally, I will produce a 25000 word report to discuss my methodology, research into the problem, execution, and evaluation of the project. These will get completed as they do on the initial 2 Gantt Charts, indicating the algorithm will be done by week 5 with learning completed by week 8 accompanied by graph generation methods by week 7. The report should be finished by week 9 with 3 flex weeks to work with.