UNIVATES UNIVERSITY

CENTER FOR EXACT AND TECHNOLOGICAL SCIENCES

SOFTWARE ENGINEERING PROGRAM

LEANDRO VIAN

Evaluating the application of Reinforcement Learning algorithms on video games

Lajeado

July 2018

LEANDRO VIAN

Evaluating the application of Reinforcement Learning algorithms on video games

Final undergraduate work presented to the Center for Exact and Technological Sciences of UNIVATES University, as part of the requirements for obtaining a bachelor's degree in Software Engineering.
Area of concentration: Artificial Intelligence

Advisor: Marcelo de Gomensoro Malheiros

Lajeado

July 2018

To my mother Iracema

ACKNOWLEDGEMENTS

To my parents Juarez and Iracema, who worked hard to give our family a better life and made me the person I am today. To my mother, I thank all the love and care she gave me during her life, and to my father, I thank for all his wisdom, friendship and continuous care. I love you both very much.

To my wife and son, Dauana for staying at my side on all occasions and giving me a son and Rômulo for being the best thing in my life and my reason to always try to be a better man. You both are incredible, I love you.

To all my family members with who I share so many good moments and helped me to get through the bad ones, thank you very much.

I would like to thank my Advisor Marcelo de Gomensoro Malheiros which helped me during the whole work, offering invaluable insight and support, this work would not have been the same without his help.

I also thank Univates and all the professors I was a student, it has been a very long ride and they all contributed to that goal, I leave Univates as a better person in all aspects.

Finally, to all those who I have met and contributed somehow to this achievement.

ABSTRACT

Artificial Intelligence has become part of our everyday for quite some time now: movies have portrayed it in its histories, news have reported of its advancements and we have seen its results in our electronics and machinery. In the latest years a new term started to gain traction, Machine Learning, with many articles, companies and media covering it, opening possibilities of what could be achieved with the ability to train computers using all the data generated nowadays. This work gives an overview of a few current Machine Learning techniques, aiming in the application of automated video game playing. In particular, it uses the Starcraft II Reinforcement Environment as a testbed for evaluating the selected automated learning strategies.

Keywords: Artificial Intelligence, Machine Learning, neural networks, Reinforcement Learning, Starcraft II.

RESUMO

A Inteligência Artificial já faz parte do nosso cotidiano há tempos: os filmes a retrataram em suas histórias, as notícias relataram seus avanços e vemos seus resultados em nossos equipamentos eletrônicos. Nos últimos anos, um novo termo começou a ganhar força, Aprendizado de Máquina, com muitos artigos, empresas e mídia o cobrindo, abrindo possibilidades sobre o que poderia ser alcançado com a capacidade de treinar computadores usando os dados gerados hoje em dia. Este trabalho dá uma visão geral sobre várias das atuais técnicas de Aprendizado de Máquina, visando sua aplicação no jogo automático de jogos eletrônicos. Em particular, utiliza-se o Ambiente de Reforço Starcraft II como ambiente de testes para a avaliação das estratégias de aprendizagem automatizadas selecionadas.

Palavras-chave: Inteligência Artificial, Aprendizado de Máquina, redes neurais, Aprendizagem por Reforço, Starcraft II.

SUMMARY

LIST OF IMAGES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
ML	Machine Learning
RL	Reinforcement Learning
SC2	Starcraft II
SC2LE	Starcraft II Learning Environment

1 INTRODUCTION

Artificial intelligence (AI) has become part of our life, being a common term in any conversation that bring up computers or software being used in the process of automate decision making areas that were previously done by humans.

After AI achieved such notoriety and its improvements were accepted as invaluable in many areas, researchers started to focus on how to construct better AIs. It became clear that many avenues could be pursued and many techniques developed decades ago by groups of study of Machine Learning could now be revisited using modern computers and programming techniques.

1.1 Motivation

Games are as old as humankind and have evolved hand in hand with us, with the advent of computers they became complex pieces of software, allowing for a multitude of genres to be developed, such as role playing, soccer, shooters and strategy.

Computer games offer an incredible new environment for AI, with a variety of challenges that need to be understood and overcome in order to achieve victory. This test chambers can help us to overcome challenges that would otherwise be too complex or expensive to be solvable directly in a real-world environment.

Currently, even if we were to limit our scope to the data being generated in the world, to the ones generated by the human interaction with electronics of all kinds, we would have amounts of data far beyond the processing power of any conventional data analysis software, even if used in conjunction with super computers.

Adding to that equation, data generated by humans through the use of applications like Whatsapp, Facebook and Instagram would turn the task of analyze such amount of data impossible with the hardware available today or for many decades to come (even without considering the fact that the amount of data generated tends to increase as we adopt more technology in our lives).

Taking into account the amount of data available and the test chamber that games offer, a new environment was created to help close the gap between games and research:

SC2LE (Starcraft II Learning Environment), created in partnership by DeepMind and Blizzard Entertainment.

An AI agent playing Starcraft II through SC2LE have to:

- Control hundreds of units in cohesion, using strategy to exert influence and secure areas on the map;
- Manage resources required to construct a base and build units;
- Make decisions based on imprecise information offered by the game screen and mini-map, both which have a “fog-of-war” effect that prevents vision if no friendly unit is nearby an area;
- Maps are big and diverse, offering many obstructions and the concept of land and air units; and
- Games may last for many minutes and decisions may not show results or consequences until later on.

All points considered, SC2 games are a difficult challenge towards building better AI tools and techniques that can be later applied to many areas, raising the simulation aspect of games to a new standard and importance.

The befits that can be achieved by merging the amount of data available with new discoveries in Reinforcement Learning (RL) are vast, most areas can benefit if some manner by either have tasks being automated or simply by making better decisions through their processes. One example is in the medical area, where images, laboratory results and symptoms can go through AI systems that have been trained with millions of cases and will help doctors make decisions or maybe a simpler case where trained AI systems can help companies run in a more efficient way, lowering costs and improving results.

1.2 Objectives

The main objective of this work is to analyze a few Reinforcement Learning algorithms and techniques, by creating agents that play the two Starcraft II mini-games. Data from running two learning algorithms in different configurations is analyzed, giving basis this work results and conclusions.

Secondary objectives, of this work are:

- Analyze the current state-of-art of Reinforcement Learning;
- Briefly analyze different Reinforcement Learning algorithms and strategies current; being used by the academy and industry;
- Analyze SC2LE and evaluate the tools presented by the library;
- Successfully Run Starcraft II agents using the selected algorithms;
- Evaluate the obtained results.

1.3 Methodology

The research methodology began with a bibliographic revision that was used as the foundation to the work, followed by data related to Reinforcement Leaning (RL) algorithms, libraries and frameworks commonly used in researches related to the subject.

With a strong base of knowledge at hand, tests were conducted using the SC2LE, where the agents were trained using the selected RL algorithms. The agents played the selected mini-games available within the tool and all data created was collected and used to help with final conclusions.

1.4 Text Overview

For best comprehension, this work is divided in chapters with the following order of presentation:

Chapter 2 presents the bibliographic revision from published material, encompassing the beginning of Artificial Intelligence and its evolution, modern-day advancements provided by AI in many fields, Machine Learning, its groups of study and algorithmic differences. Moreover, we also address Reinforcement Learning and its need for data, games, its electronic segment and how AI affects it. The Chapter 3 presents all the technical steps involved with the setup of the game and mini-games inside SC2LE, details about the communication interface between game and library, agents and the RL algorithms and software libraries used. In Chapter 4 the experiments conducted are described, followed by the presentation of the data collected during the tests. The results obtained are then analyzed. Chapter 5 concludes with some final remarks and possibilities for future work.

2 LITERATURE REVIEW

This section presents and analyzes the theoretical references used during the development of this work. Each section will address a specific aspect or subject important to the whole process, being discoveries that led to the current state of the knowledge and technology or specific aspects that will be used and need to be well understood in order for them to be well handled.

2.1 Beginnings of Artificial Intelligence

Computing History is marked with Alan Turing's conception of the "Turing machines" and the release of his paper "Theory of Computation", which is the cornerstone to the computer revolution that followed since then.

As presented by (RUSSEL et al, 1995), the first work that is now generally recognized as AI was done by Warren McCulloch and Walter Pitts, in 1943, when they created a model of artificial neurons that were based on three concepts. The first was the physiology and function of neurons, the second was propositional logic and the third was the Turing's theory of computation.

With that concept in hand, they went ahead to show that any computable function could be computed by a network of said neurons. Soon after Donal Heeb (1949) gave his contribution by demonstrating an updating rule that would allow the neurons to learn, by allowing the strength of connections to be modified.

In 1951 the first neural network computer called SNARC was built by Marvin Minsky and Dean Edmonds in the Princeton mathematics department, all the while the first chess playing programs were already being written by Claude Shannon (1950) and Alan Turing (1953).

As the result of those first steps, Dartmouth College hosted a two month workshop during the summer of 1956, where some programs were presented, including a reasoning program called the Logic Theorist from Alien Newell and Herbert Simon. Even tough the workshop did not result in any major advancement, it was there that the term Artificial Intelligence (AI) was agreed to be adopted as the field name.

2.2 Evolution of AI

In the first decade of the advent of AI, a lot of expectation was created and much enthusiasm could be seen around the subject as pointed out by (RUSSEL et al, 1995): “Given the primitive computers and programming tools of the time, and the fact that only a few years earlier computers were seen as things that could do arithmetic and no more, it was astonishing whenever a computer did anything remotely clever.”

During this period, the first program to implement a human like approach to solving puzzles was created by Newell and Simon, it is considered the first one to embody the thinking humanly approach by mimicking the way we consider subgoals and possible actions that could be taking.

During this decade, the LISP high level language was created by John McCarthy, witch came to be the dominant AI programming language and is still in use nowadays. McCarthy also helped to develop time-sharing minicomputers and published a paper in which he describes the first complete AI system to utilize general knowledge of the world to offer solutions.

During 1970, AI started to encountered its first real issues, as the predictions of coming successes came to fail. Problems such as the lack of knowledge by the programs about the problem being solved were noticeable, such as a translation problem presented by (RUSSEL et al, 1995) “The famous retranslation of ‘the spirit is willing but the flesh is weak’ as ‘the vodka is good but the meat is rotten’ illustrates the difficulties encountered.”

Another issue encountered was the fact that many problems being tackled by AI were hard to present in a manageable way in order to be solved. Before the NP-completeness theory, it was thought that with enough processing power and memory, any problem could be solved, which proved false when theorems with more than a dozen facts could not be solved.

A third problem was presented as the fact that the basic structures that were considered the base for the construction of intelligent behavior, could in fact learn anything that they could represent, but managed to represent very little. This lead to an almost full stop in the research of multilayer networks, beginning only to recover in the 80's.

This lead to the concept of knowledge-based systems, while the initial AI research focused on solving general problems putting together elementary reasoning steps to arrive at the desired solution, it proved to offer less then desirable results for more complex problems.

Later on, research went to the use of more complex reasoning steps and more suited knowledge bases, which offered better results but were too specific, as told by (RUSSEL et al, 1995): “One might say that to solve a hard problem, you almost have to know the answer already.”

As mentioned before, the 80's presented a real comeback to the AI research and the rise of the first AI based industries, rising to notoriety offering the world the first successful commercial expert system, R1. On par with that, the Japanese presented a ten year plan to build intelligent computers, which led the United States and Britain to do the same in fear of the Japanese dominion.

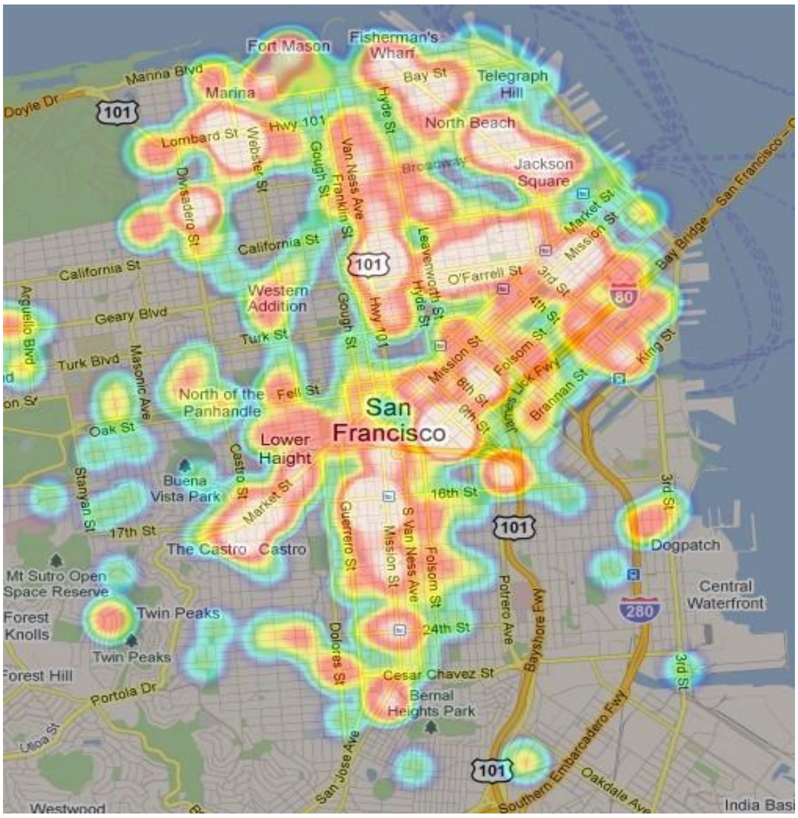
2.3 Recent AI

In the last two decades AI has evolved a tremendous amount, stopping from being something that would be seen of specific circles of talk or science fiction movies, to becoming part our our daily routine. Things such as personal voice assistants, self-driving vehicles or even behavioral suggestive algorithms started to permeate all aspects of our life, as pointed out by (ADAMS, 2017): “The machines haven't taken over. Not yet at least. However, they are seeping their way into our lives, affecting how we live, work and entertain ourselves.”

Some interesting advancements can be pointed out, such as the one that resulted in the creation of new business model that competes with the traditional taxi companies. The Uber company offers their clients the ability to ask for a ride knowing exactly how much they will have to pay, and how long will take for the car to arrive and for the trip to be complete. This is done with the use of a AI system that incorporate specific behaviors that calculates distances, car availability, traffic and demand during specific times of the day.

A representation of a heat map (TECHEMERGENCE, 2017) used as input by the Uber system to help calculate tariffs can be seen in Figure 1.

Figure 1: Heatmap used by Uber AI to calculate costs, availability and time.



Source: (TECHEMERGENCE, 2017)

Another example of the regular use of AI in our routines be can be seen in the airline companies, which uses AI autopilots as a common tool and as much as possible, to many passengers surprise as noted by (NARULA, G. 2017): “The New York Times reports that the average flight of a Boeing plane involves only seven minutes of human-steered flight, which is typically reserved only for takeoff and landing.”

Following the airline examples, another venue has received a lot of attention, the self-driving cars one, with large companies behind it, such as Google and Tesla. Tesla approached the problem with his AI using the cars sensors to make decisions based on whats was pre determined by the company, while Google went a bit farther, leaving its AI to learn on its own in hopes that it will learn how to drive as we do, trough experience (TECHEMERGENCE, 2017).

The list of examples is vast and exist on most fields, digital media for example, show the use of AI on video games by making better adversaries and challenges, purchase prediction, helping clients to “find” what they might want and fraud detection helping e-commerce owners and banks (TECHEMERGENCE, 2017).

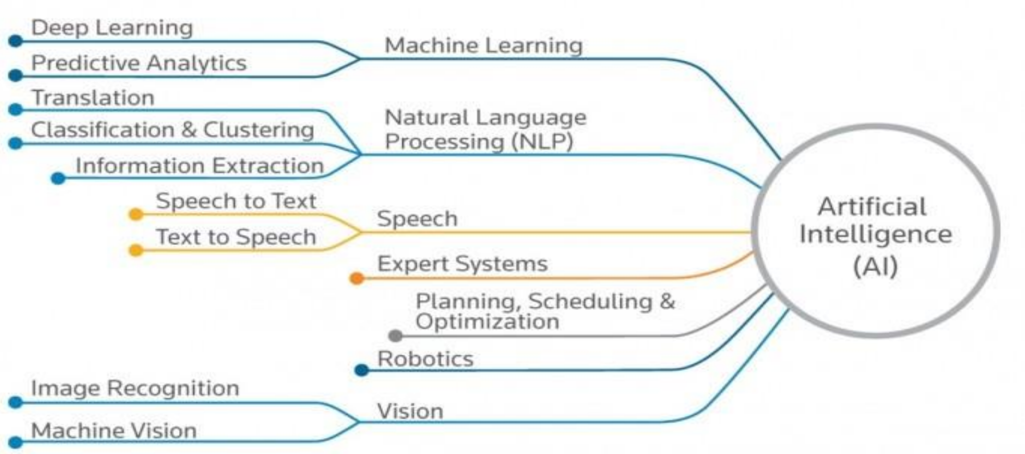
Security surveillance in addition to cameras and monitor, now offer monitoring systems that analyze the video feeds and using trained security algorithms can determine if the movement detected is a threat and warn a human responsible. Similar technologies are implemented within “smart homes”, which have many interconnected technologies that work in cohesion to offer the owner an experience tailored for him (TECHEMERGENCE, 2017).

Areas improved by AI are many, researches are on full speed and opportunities are vast, future predictions only seem to increase the expectations and as pointed out by (ALBRIGHT, D. 2017): “… sometimes it’s obvious what its’ doing, like when you ask Siri to get you directions to the nearest gas station. Sometimes it’s less obvious, like when you make an abnormal purchase on your credit card and don’t get a fraud alert from your bank. AI is everywhere, and it’s making a huge difference in our lives every day.”

Despite the numerous advancements and researches released making huge headlines on news around the Internet, most of the time they are referred as AI for the sake of simplicity and to help readers associate the information with the overall area of expertise. In fact AI is a huge area of study which encompass many sub-fields, each with its sub fields, many being responsible for leading the way to many discoveries in the AI field.

A representation of AI and its sub-fields can be seen in Figure 2.

Figure 2: Breakdown of the AI field and its sub areas.



Source: (LINKEDIN AI BLOG POST, 2017)

With the amount of AI sub-fields making discoveries that draw attention, a lot of investment is being made towards more research, which tend to result in an even bigger permeation of AI in our lives, as pointed out by (ATTICK, R 2016): “The proliferation intelligent technology is resulting in advanced machines and systems which are capable of sensing, thinking, reasoning, finding patterns, predicting, communicating and acting faster than humans ever dreamed possible.”

2.3.1 Machine Learning

Machine Learning is a field that is attracting a lot of currently, being responsible for many new advancements, the term was coined by Arthur Samuel in 1959 to whom was attributed the creation of the world first self-learning program. The program he developed played checkers and used a search tree of the board to determine the possible moves based on the state of the board (RUSSEL, 1995).

The idea behind ML is to have the program teach itself by iterating its logic using all the data available t him as input, each iteration results and some knowledge that will help it build its neural network. This process was described by (SAMUEL, 1959): “Machine learning is the sub-field of computer science that ‘gives computers the ability to learn without being explicitly programmed’.”

Modern society produces an incredible amount of data each day, being it originated from smartphones and computers, applications and social media or even antennas and satellites. With an unending stream of data to be feed, ML systems are discovering new ways of sell and advertise products, helping business owners to target audiences, helping doctors making decisions and even helping with the test of new medicines (DOMINGOS, 2015).

A recent example of ML being used to help solve problems was presented in an article by (KUBAT et al, 1995): “In this paper we describe an application of machine learning to an important environmental problem: detection of oil spills from radar images of the sea surface”, which resulted in the conduction of field tests using the information learned by the ML system as guide.

2.3.2 Vision

Object recognition or Computer Vision is another field under the AI category which involves the process of taking a picture, movie or stream and interpreting it so computers can understand whats been seen and use that information as accordingly.

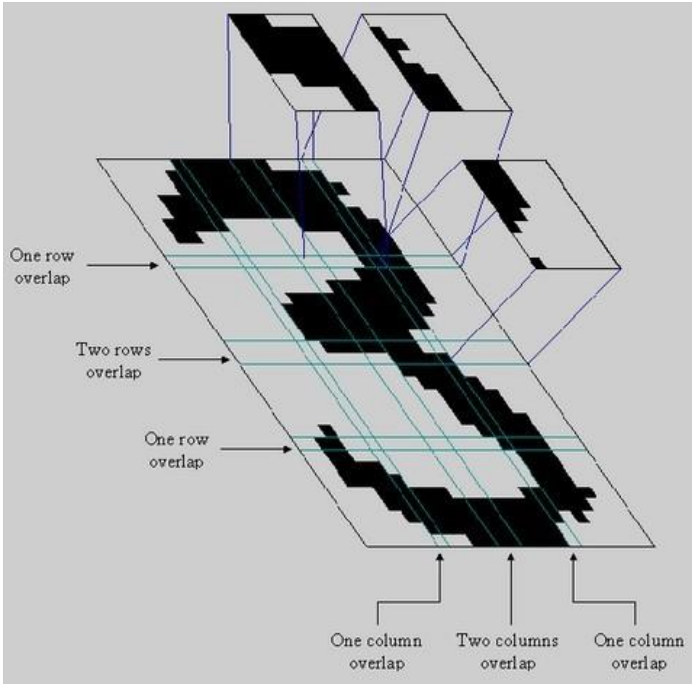
A task that is natural to humans, looks so simple and is done almost instantaneous, actually is a real complex system as described by (SONKA, 2014): “When a human tries to understand an image then previous knowledge and experience is brought to the current observation. Human ability to reason allows representation of long-gathered knowledge, and its use to solve new problems.”

For machines the same task is way more difficult and involves complex steps and integration of systems in order to translate what is being seen to a computational model. These steps are normally broken out as the follow:

- Image Acquisition: is the process of producing the images itself, using a camera to creating the image data in pixels that is read by computers.
- Preprocessing: before the image can be used by the computer it must first be processed to ensure the image will satisfy certain requirements, some of those process are re-sampling, noise reduction and contrast enhancement.
- Feature extraction: in this step various aspects of the image are analyzed and taken in consideration, such as lines, edges and points of interest.
- Detection/segmentation: at this point the image is broken down and the parts considered most relevant will be taken to the next stages, chosen parts normally contain a specific interest point or many grouped interest points.
- High-level processing: at this step, the chosen part from segmentation are analyzed and go trough some processes such as image recognition, image registration, estimation of specific parameters and data assumptions.
- Decision making: lastly is the step deciding if the if the image accepted by the application, using tests as automatic inspections, recognition matches and flags for human reviews.

A depiction of the process of breaking down an image into usable information required for the analysis and classification of images can be seen in Figure 3.

Figure 3: Example of image breakdown for analysis



Source: (SONKA, 2014)

2.3.3 Robotics

The term ‘robot’ was coined by the writer Karel Capek, is was defined by the Robot Institute of America in 1979 as “A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks.”

As it has happened in many areas, robotics saw may advancements in its area due the the use and implementation of ML techniques, being it during manufacturing of robots, testing or software development. This increase trend was is pointed out by (FAGGELLA, 2016): “Like many innovative technological fields today, robotics has and is being influenced and in some directions steered by machine learning technologies.”

One such example of the use of ML techniques within robotics can be seen in the medical area, where assistive robots are being use to sense and process sensory information that allows them to performs actions that help seniors and people with disabilities.

Another good example that has attracted a lot of attention lately are the warehouses assisted by robots, where humans only select the client order to be collected and the robots go around the warehouse picking boxes that contain the items asked for. Those robots rely on many sensors and cameras that allow them to navigate complex mazes, pick and stack heavy boxes and bring them to the desired location, all that without crashing with anything (FAGGELLA, 2016).

As pointed out by (FAGGELLA, 2016), “The slight difference between the two may be in kinematics as applied to robot vision, which encompasses reference frame calibration and a robot’s ability to physically affect its environment.”, making the process more complex than regular computer vision.

2.3.4 Speech Recognition

Speech recognition is not new to our society, we have had some machines and computers with that characteristic for decades, but only recently with the advent of ML it was possible for it to be used outside of specific controlled environments or scenarios, as pointed out by (FELLOW et al., 2013) “In recent years, the machine learning (ML) and automatic speech recognition (ASR) communities have had increasing influences on each other.”

The technology behind speech recognition require the following steps:

- First a audio file is created by recording a sound.
- Then the audio file is sampled and turned into numbers by analyzing its wave heights during timed periods.
- With the resulting numbers, the audio can be preprocessed and after the creation of sample groups, that allow a graph to be plotted.
- To facilitate the ML work, the sound is broken down into smaller pieces that than can be used to create a spectrogram of the sound.
- ML than works on pieces of the spectrogram to interpret the sounds made, learn, understand and offer the information required for the system to respond.

This process allowed ML to be used at its full potential and provided many benefits to the field, allowing for phones, smart watches, gadgets and automated homes to perfectly understand their owners and even adapt to them, despite their accents, voice pitch or voice volume. This incredible process was described by (GEITGEY, 2016): “For just $50, you can get an Amazon Echo Dot—a magic box that allows you to order pizza, get a weather report or even buy trash bags—just by speaking outloud.”

2.3.5 Expert Systems

For many decades expert systems have been used in many areas to help solve complex problems, its first architecture was created using basic programming rules such as if-else and procedural coding. The most interesting aspect of expert systems is the fact that it allows for programs to be used by people with almost no understanding of coding, due its the information used being explicit instead of implicit in the code.

After a phase of acceptance and widespread use and recognition of its benefits, expert systems were called out by its limitations, this need limitations pointed out for the need of renewal in the field as pointed out (SELFRIDGE et al, 1987): “We believe that these next-generation expert systems will have to be based on cognitive models of expert human reasoning and learning in order to perform with the ability of a human expert.”

In recent years some signs of innovation were show as Machine Learning was introduced and some aspects of expert systems were than adjusted and helped create new segments such as recommendation systems.

2.3.6 Natural Language Processing

Natural Language Processing (NLP) has been studied for over 50 years and became a field by itself with the advent of computers. NLP is defined as the study of automatic manipulation and understanding of natural languages evolved by man.

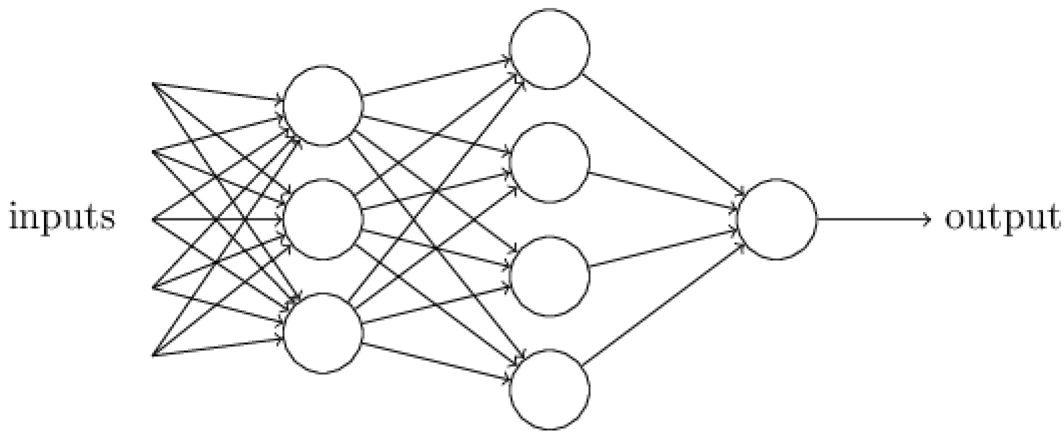
One example of the use of Machine Learning in partnership with NLP can be seen in the sentiment analysis conducted on movie reviews posted by users on the website Rotten Tomatoes. The experience conducted in the website resulted in some expressive results related to sentence analysis as demonstrated by (IYYER et al, 2015): “We introduce a deep unordered model that obtains near state-of-the-art accuracies on a variety of sentence and document-level tasks with just minutes of training time on an average laptop computer.”

Another good example comes from the machine translation segment within NLP, where a computer is responsible for converting a source text from one language to another. A research paper which translated texts from English to French using ML techniques presented the following results: “… we obtained a BLEU score of 34.81 by directly extracting translations from an ensemble of 5 deep LSTMs (...). This is by far the best result achieved by direct translation with large neural networks” (SUTSKEVER et. al, 2014).

2.3.7 Neural Networks

Neural networks is a paradigm for processing information, inspired by the way our nervous system works. Just as in our brain contains neurons that are connected and work together to process everything we need, so do neural networks, where its architecture is made of interconnected neuron models in a way they can pass and receive information from one to another, as can be seen in Figure 4.

Figure 4: Example of neural network model



Source: (NIELSEN,2017)

The first concept of an artificial neuron was created during the 1950 and 1960 period by the scientist Frank Rosenblatt, inspired by earlier works from Warren McCulloch and

Walter Pitts. That model consisted artificial neuron model called Perceptron, who received of multiple binary inputs and produced a single binary output (NIELSEN, 2017).

The most accepted artificial neuron model used currently is the sigmoid neuron, who can output values between 0 and 1, based on the result of a sigmoid function calculated using all the input values received by the neuron. That change in the sigmoid neurons solved the problem of value flipping that happened with perceptrons as state by (NIELSEN, 2017): "... a small change in the weights or bias of any single perceptron in the network can sometimes cause the output of that perceptron to completely flip, say from 0 to 1."

A common use of Machine Learning for teaching systems can be seen in the handwriting recognition field, where systems responsible for recognition are trained with ML and easy reach levels of accuracy of 99% or more with state-of-art works. This allows for such systems to be so reliable that even banks use them to validate checks and documents that require user signatures (NIELSEN, 2017).

2.4 Machine Learning groups of study

Machine Learning has been studied for many decades, and during all those years many groups appeared, offering different ways of implementing ML systems, based on field of science they were most close to, as pointed out by (DOMINGOS, 2015): "The search for the master algorithm is difficult but also stimulated by the rival schools of thought that exist in the area of machine learning."

The Symbologists tackled the ML problem with the idea that all intelligence can be reduced to the manipulation of symbols, much as am mathematical equation is solved by replacing expressions. This group believes that some preexisting knowledge is necessary prior to learning and for that mean use inverse deduction to achieve their solutions.

The Connectionists believe that imitate learning they must look at how our brains work, so as the brain works by adjusting neuron connection strength, so does their implementation. This group works by comparing the exit value with the desired value and adjust its connections accordingly, their solution is called backpropagation.

The Evolutionaries inspire themselves in how the planet evolved, and for that they believe that the best way to learn is by using a natural selection approach, to solve a structure of learning. This groups developed a method called genetic programming that works by making small changes in values and checking if a better one is found and then used as the default value.

The Bayesians understand that for this problem to be solved, the most important aspect that need to be solved is the uncertainty, because all learning is uncertain and need to be tested before it can be considered correct. The solution created by this group is called probabilistic inference which allow them to incorporate new evidences in the data used as quickly and efficient as possible.

The last group are the Analogizers, they believe that the best way to handle learning is by recognizing similarities between situations, from there they can infer other similarities. Their bigged problem is to how they can determine similarities, for that they developed the kernel machines, which work by determining which memories experiences should be remembered and how to combine them with new ones.

Each group tackled the ML problem with a different perspective, has different difficulties and offer different ways to solve problems, they even fought each other at times, but their combined work allowed the ML community to the level of importance they are today, as can be seen in Table 1.

Table 1: Machine Learning groups information

Group	**Origins**	**Master Algorithm**
Symbologists	Logic, philosophy	Inverse deduction
Connectionists	Neuroscience	Backpropagation
Evolutionaries	Evolutionary biology	Genetic programming
Bayesians	Statistics	Probabilistic inference
Analogizers	Psychology	Kernel machines

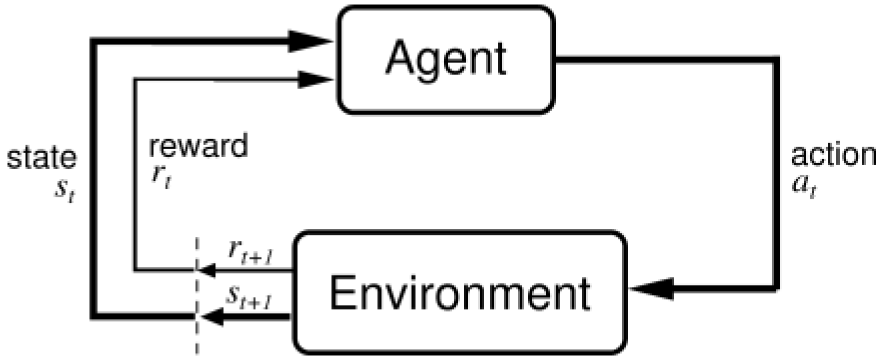
Source: (DOMINGOS, 2015)

2.5 Reinforcement Learning

Reinforcement Learning (RL) is a sub area of Machine Learning that is focused on how to teach agents the best actions to take on a specific scenarios, this is done by letting the agent take an action, than analyzing the outcome and comparing that to what is considered a good outcome. The idea is that with each cycle of action, reward and analyzes, the agent will gradually understand what actions produce the best outcomes and will steer toward those actions.

In Figure 5 we can see the basic representation of process agents go trough during their cycle of action, rewards and analyzes.

Figure 5: Representation of agent behavior



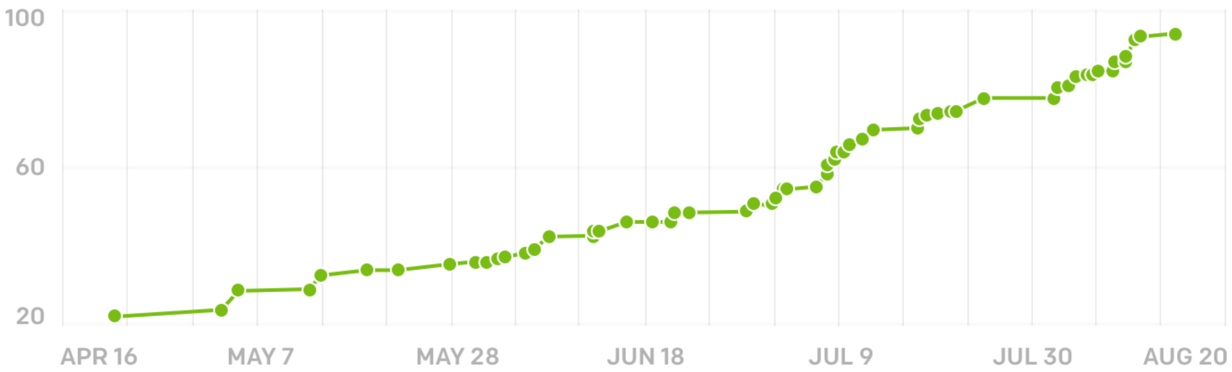
Source: (SHAIKH, 2017)

Reinforcement Learning was responsible for many great feasts in the latest years, being considered a buzzword in the IT community, this influx of interest also attracted many new followers, researchers and investors to the area, looking forward to new discoveries and opportunities (SHAIKH, 2017) .

One examples of RL being at the spotlight was seen during a professional gaming tournament called The International, where team of players fought each other competing for prizes of a total of $24,787,916. During the tournament a presentation was made, putting a agent trained with RL to play Dota 2 against the best Dota 2 players in the world, the result left the crowd ecstatic as the agent won easily against all human opponents (SHAIKH, 2017) . The agent was created by the non profit research company OpenAI, which after developing the agent, trained it for over four months against itself in countless games where he learning on its own how to play game. During those four months the bot went from completely clueless agent that didn’t knew how to navigate the map, to being able to beat the top world player handily (OpenAI, 2017).

In Figure 6 we can see the evolution of the agent match making rating (MMR) evolution during the 4 month training period prior to the tournament. The timeline represents that 15% of players are below 1.5k MMR, 58% of players are below 3k and 99% are below 7.5k (OpenAI, 2017).

Figure 6: Agent evolution graph.



Source: (OPENAI BLOG POST, 2017)

2.6 Games

Games are a normal part of our society and history suggest that it has been so for thousands of years, but nowadays are commonly known and used just as a form of entertainment, where in the beginning it had a much more important role.

During the early days of our history games had a vital role within communities, they helped people pass time, but not like today, they literally helped people to relax and forget at least for a while problems like lack of food, thirsty, health problems or dangers.

Some of the oldest dated games that we have evidence are thousands of year old and help us understand how important games were in the human culture, as said by (KUMAR et al, 2017): “Human history and games are inextricably intertwined. Irrefutable evidence resounds down through the ages that fun and games are not frivolous pursuits per see, instead, they come naturally to us as essential parts of being alive.”

Board games are one of the most common games we play, one of the oldest board game found is dated over five thousand years old, the rules are not known but it was played by the Egyptians. In Figure 7 we can see a depiction made by old societies of games being played.

Figure 7: Depiction of old societies playing games

Source: (KUMAR et al, 2017)

Another old example of the game the endured till today are the dice, who are so common and used as the main game or as a tool for of a plethora of other games. Dice are dated with with more than three thousand years old, as pointed by (KUMAR et al, 2017): "… archaeologists discovered a 3,000-year-old set of dice! We don't know exactly what games those early Persians would have played with them, but the popularity of dice has endured throughout the centuries."

2.6.1 Electronic games

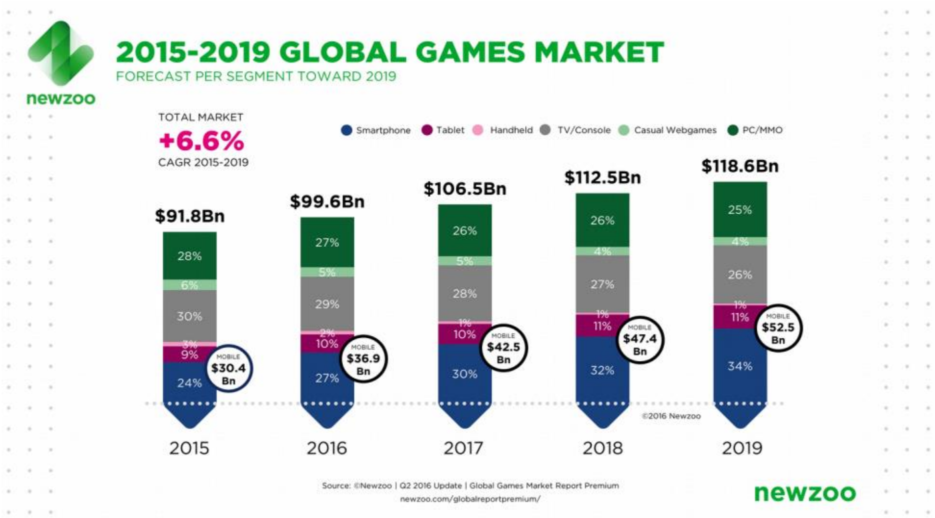
Electronic games began its history back in 1950 with the creation of the first "machines" that allowed people to interact with it and play a tic-tac-toe game, followed by one that resembled a kind of 2D tennis match. But the first game created and sold to be played in a "video-game" attached to a television was Pong in the 70s, who then helped electronics games to spread in the US (KAPLAN, 2013).

Following the creation of video-games, the arcades were created where players could spend a coin to play in the machine, creating with that stores that were social hubs of for many people and were many friendships were created with the help of games. In the arcade genre the first success game was Space Invaders, which was a huge success in the US and Japan as stated by (KAPLAN, 2013): “In two different countries on opposite sides of the globe, Japanese and American teenagers, although they could not speak to one another, were having the same experiences thanks to a video game.”

Personal computers started to appear during the 70's, with the models that helped the development of games appearing in 1977 with the Commodore, Apple II and TRS-80. With those new types of computers, games could become a bit more complex, using a little bit more processing power and memory, offering good results as stated by (KAPLAN, 2013): “The development of video games for the personal computer platform expanded the ability of video games to act as media by allowing complex stories to be told and new forms of interaction to take place between players.”

During the latest decades much has evolved in the game industry, computer became more powerful and so did consoles, even sprouting a “console war” with companies fighting to have the best games, console and most faithful followers. More recently new medias for games appeared, with smartphones and tablets having such good hardware as to be able to run modern titles with good performance and graphics, thus creating a new segment in the industry (KAPLAN, 2013).

Figure 8: Global market for games



Source: (NEWZOO Website, 2017).

Electronic games are a huge segment in the modern society, being it analyzed trough a commercial, social or creative perspective, it moves billions of dollars every year, influence whole generations of players and spread incredible narratives. With a market so big that shadows the movie and music industry, the importance of gaming can be seen in Figure 8, which shows global estimates (NEWZOO, 2017).

2.6.2 AI in games

Artificial Intelligence is an area inside the game industry responsible for making the games look smart, being it with enemies that behave with human nature, make reasonable pathfinding decisions or simply choose the best tool/weapon for a specific situation (MILLINGTON, 2009).

All games have some sort of AI responsible for all manner of jobs, but prior to 1990 all of them used similar techniques which relied on hard coded choices based on if-else decisions that determined the next action. During 90's games did still relied on defined states but started to introduce AI improving techniques, such as sense simulation, which allowed enemies to notice things such as dead friends and react accordingly.

During the next years other advancements were made, such as strategy games introducing noticeable AI pathfinding for units and formation motion for groups, while in the 2000's some games presented things such as neural network-based units. Some other topics still were not solved, like RPGs using tree-based dialog systems or sports games having trouble with some dynamically calculation required for sports simulation (MILLINGTON, 2009).

While the need for complex AI depends on each game, with some requiring an advanced AI to be enjoyable, others don’t see much improvement by implementing a complex system, relying on basic solutions. The common use for AI in most modern games is described by (MILLINGTON, 2009): “The AI in most modern games address three basic needs: the ability to move characters, the ability to make decisions about where to move, and the ability to think tactically or strategically.”

3 DEVELOPMENT

This section presents details relative to the tests, research, programming, and results obtained during the experimentation phase. A special focus is given to neural network paradigms and sets of algorithms and their characteristics related to the problems in discussion. Also, the tools that were used are discussed, such as Starcraft II, SC2LE, and TensorFlow. All are either at the core of the work or present significant benefits that justify their use at one of the later stages.

3.1 Implementation tools

The following tools were be used in this implementation:

- Starcraft II
- SC2LE
- TensorFlow machine intelligence library
- OpenAI Baselines for DQN and A2C implementations

3.1.1 Starcraft II

Starcraft II is the second game of the Starcraft franchise developed by Blizzard Entertainment and released in 2007. The first game of the franchise was released in 1998, being a huge success. During its period it developed a community of developers who built scripted bots to fight each other in competitions to see which was the best (DEEPMIND et al, 2017).

The second game was also considered a success, offering the following challenges to the RL agents:

- Two resources that need to be mined and managed (minerals and gas).
- Construction of production buildings.
- Different units and buildings, each with its own actions and options.

- Completely different races.
- Large maps with different terrain level.
- Land and air units.
- Training of an army.
- Management of hundreds of units during battle.
- Imperfect information due to the fog-of-war effect.
- Decisions that may only see their consequences later in the game.

With the release of SC2 the developers behind the game decided to offer tools to help the community build their bots. Following the initiatives of Deepmind, an Artificial Intelligence research company, they partnered to create the tools necessary.

Figure 9 is possible to see a Starcraft II game being played, showing some of its units, structures and player interface.

Figure 9: Starcraft II Gameplay

Source: (ROCKPAPERSHOTGUN STARCRAFT II REVIEW, 2017)

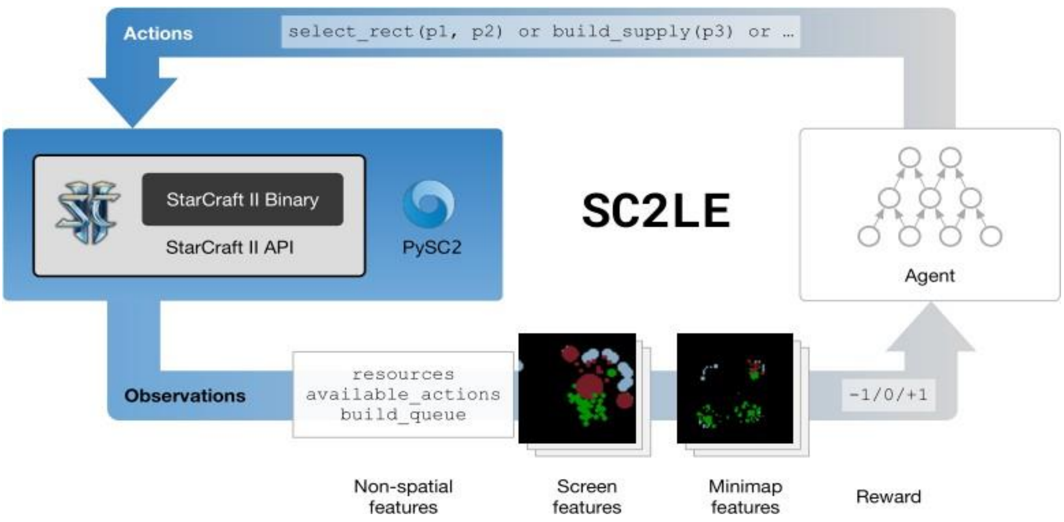
3.1.2 SC2LE

Starcraft II Learning Environment (SC2LE) is a set of tools created to allow the community of AI research using the Starcraft II game. It is composed of the following tools:

- The SC2client-proto, a Machine Learning API created by Blizzard that allows direct control of the game.
- The PySC2, a toolset written in Python that facilitates the use of the Blizzard API with the agents.
- Packs of replays from SC2 with more than 500 thousand replays to be used for RL.
- RL mini-games to help with the research of specific areas of the game, allowing for the training of specific areas of the game, removing complexity involved with the need for the agents to tackle the whole game from start.

Figure 10 shows a representation of the SC2LE components during its use by an agent, demonstrating the input, decision making, and reward analyze flow.

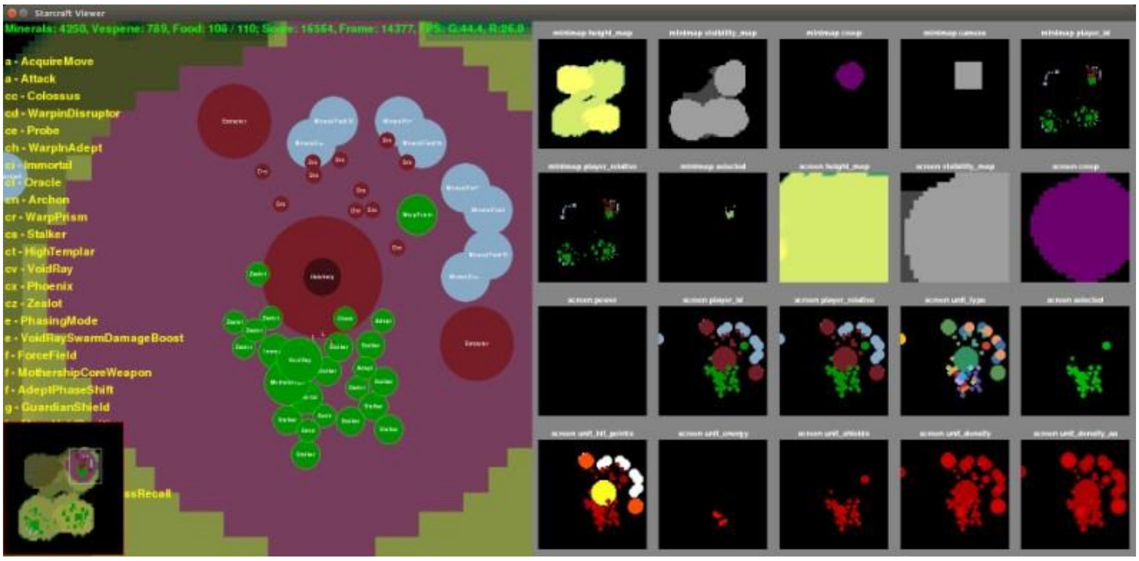
Figure 10: Representation of SC2LE flow



Source: (DEEPMIND, 2017)

Figure 11 shows a representation of the image of the game broken down into layers that related to the various kinds of information necessary for the agents to make decisions

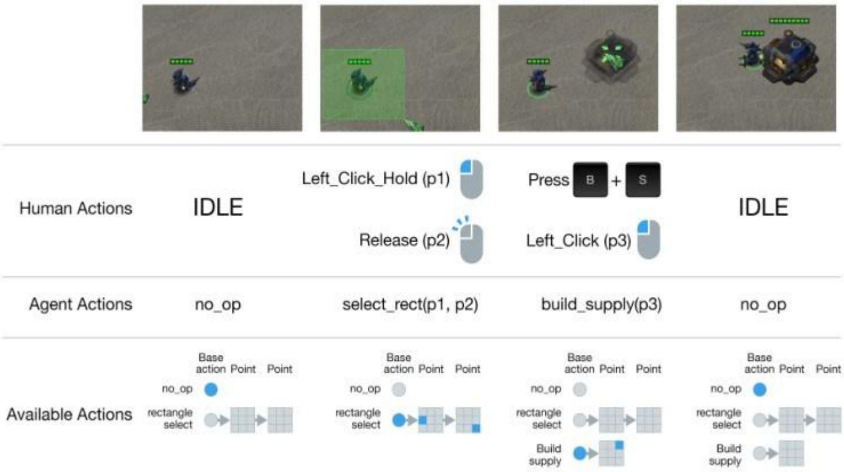
Figure 11: Depiction of the different information layers



Source: (DEEPMIND, 2017)

The process of interacting with the screen is described in the Figure 12, where a comparison of the human and agent inputs is made.

Figure 12: Comparison between the player and the agent of the inputs



Source: (DEEPMIND, 2017)

3.1.3 Agents

The SC2LE agents consist of stages that are similar no matter which mini-game or RL algorithm is being used. The process starts reading the current game state and doing an analysis of the rewards received from the last episode, which then is used in conjunction with the current neural network to choose the next action that's going to be taken.

The agent in this work was built using the Python language and packages available for its environment, such as the OpenAI Baselines. This structure allowed the agent to be created with a lean structure that only diverged when defining the RL algorithm that was selected for the agent to be run with. That is, most of the code was generic enough to provide a testbed for comparison of distinct RL algorithms.

3.1.4 Mini-games

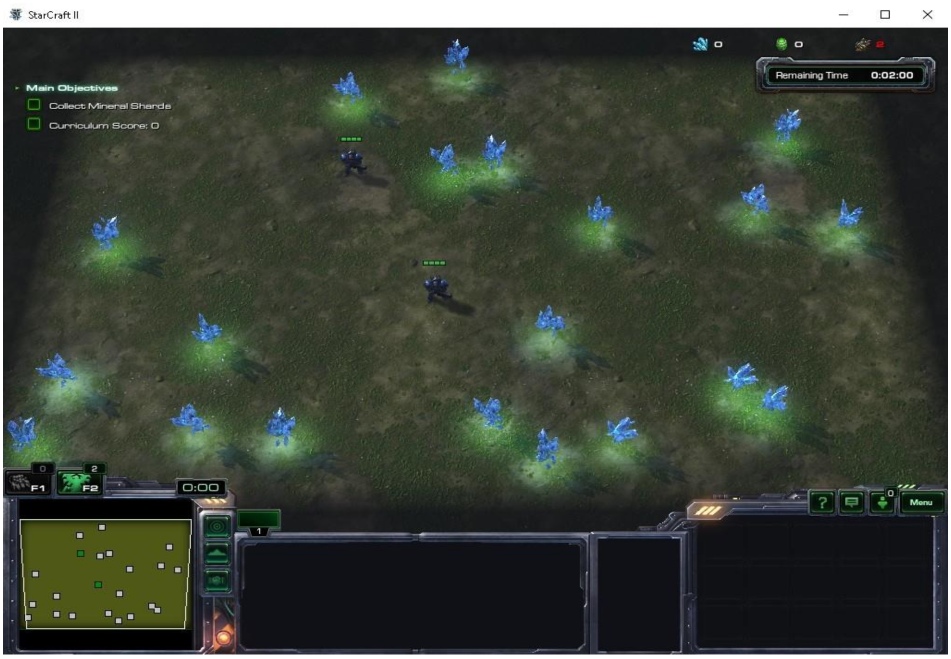
SC2LE mini-games consist of small Starcraft II maps that isolate elements of the full game, allowing for agents to be created and tested in a much simpler environment and with a smaller degree of complexity. Those mini-games break the full game into smaller tasks such as move to specific locations, collect resources such as minerals and gas, build units, defeat enemies and find the enemies themselves.

Collect Mineral Shards

The first mini-game used was the Collect Minerals, which is one of the most basic tasks required to master in order to successfully play the game. This mini-game challenges the agent to collect the game resource as efficient as possible controlling two individual units at the same time.

In this map the agent starts with two marine units and must use them to collect all 20 mineral shards available on the map, as shown in Figure 13.

Figure 13: Starting state of the mini-game



Source: AUTHOR (2018)

Each episode of the mini-game lasts for 120 seconds, time during which the agent will try to collect as much minerals as possible, if all the initial 20 mineral are collected, a new set of 20 minerals are spawned. An example of the mini-game session can be seen in Figure 14.

Figure 14: Mini-game being played

Source: AUTHOR (2018)

The final score of the agent is based on the amount of minerals collected during each episode, and can be seen on the top left corner of the mini-game screen or printed to the console, as shown in Figure 15.

Figure 15: Scores by episode

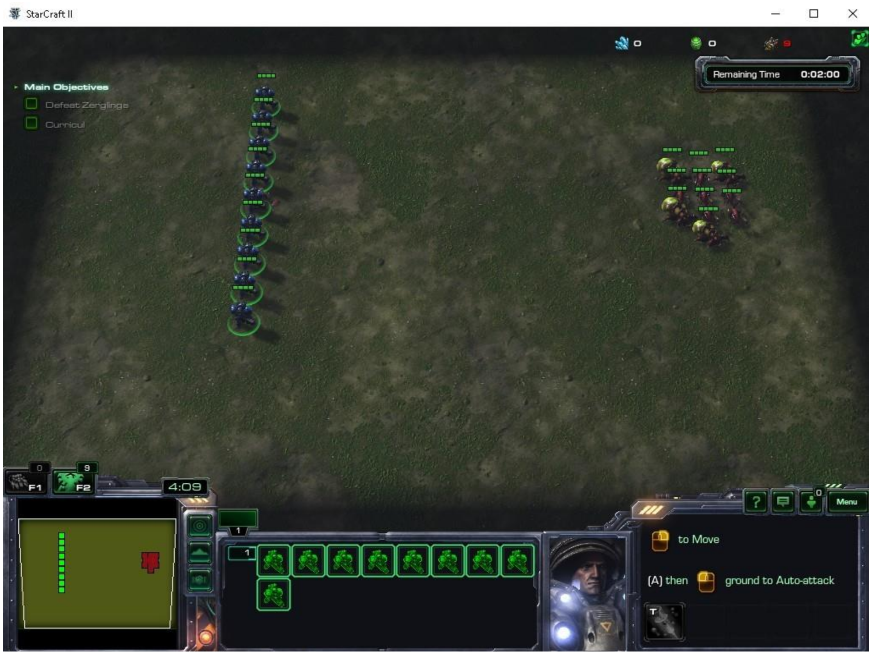
```
saving model to _files/models\temp_testing, step 0
episode 0 ended. Score 13.000000
episode 1 ended. Score 16.000000
episode 2 ended. Score 12.000000
episode 3 ended. Score 17.000000
episode 4 ended. Score 18.000000
episode 5 ended. Score 17.000000
episode 6 ended. Score 10.000000
episode 7 ended. Score 6.000000
episode 8 ended. Score 11.000000
episode 9 ended. Score 23.000000
episode 10 ended. Score 17.000000
episode 11 ended. Score 22.000000
episode 12 ended. Score 16.000000
episode 13 ended. Score 16.000000
episode 14 ended. Score 16.000000
episode 15 ended. Score 18.000000
```

Source: AUTHOR (2018)

Defeat Banelings and Zerglings

The second mini-game used was the Defeat Zerglings and Banelings, which mimics the challenge the agent with the combat mechanic from the game, requiring the agent to adeptly use the units available to defeat the enemies. In this map, the agent starts with nine marine units and must defeat six zerglings units and four banelings units, as displayed in Figure 16.

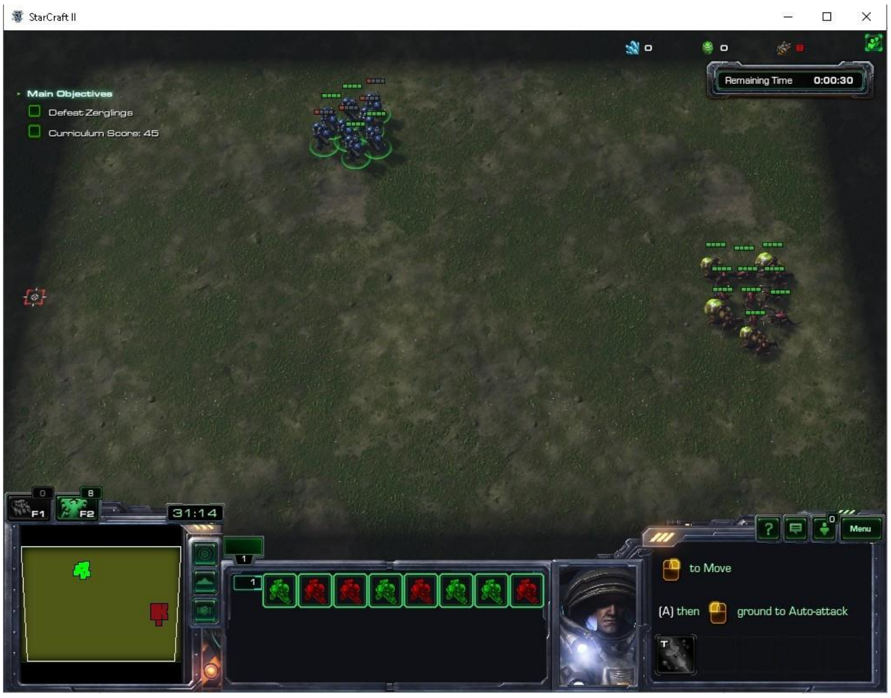
Figure 16: Mini-game being played



Source: AUTHOR (2018)

Each episode of the mini-game lasts for 120 seconds, the time during which the agent will try to kill as many enemies as possible. If all the enemies are defeated, a new set of six zerglings and four banelings are spawned and the agent receives four extra marine units at full health. An example of the mini-game session can be seen in Figure 17.

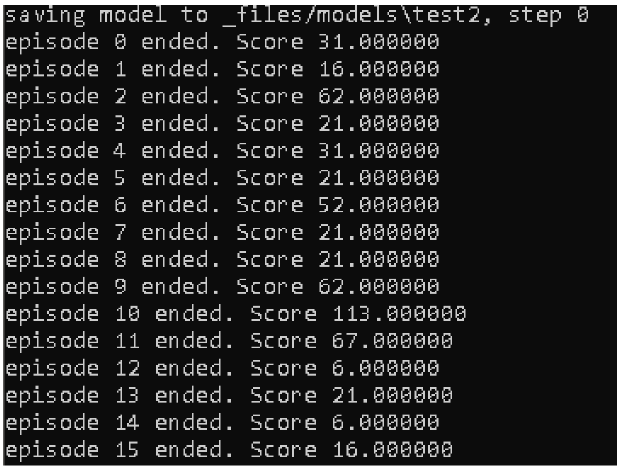
Figure 17: Mini-game being played



Source: AUTHOR (2018)

The final score of the agent is based on the amount of enemies killed during the episode, and can be seen on the top left corner of the mini-game screen or printed to the console, as shown in Figure 18.

Figure 18: Mini-game being played



```
saving model to _files/models\test2, step 0
episode 0 ended. Score 31.000000
episode 1 ended. Score 16.000000
episode 2 ended. Score 62.000000
episode 3 ended. Score 21.000000
episode 4 ended. Score 31.000000
episode 5 ended. Score 21.000000
episode 6 ended. Score 52.000000
episode 7 ended. Score 21.000000
episode 8 ended. Score 21.000000
episode 9 ended. Score 62.000000
episode 10 ended. Score 113.000000
episode 11 ended. Score 67.000000
episode 12 ended. Score 6.000000
episode 13 ended. Score 21.000000
episode 14 ended. Score 6.000000
episode 15 ended. Score 16.000000
```

Source: AUTHOR (2018)

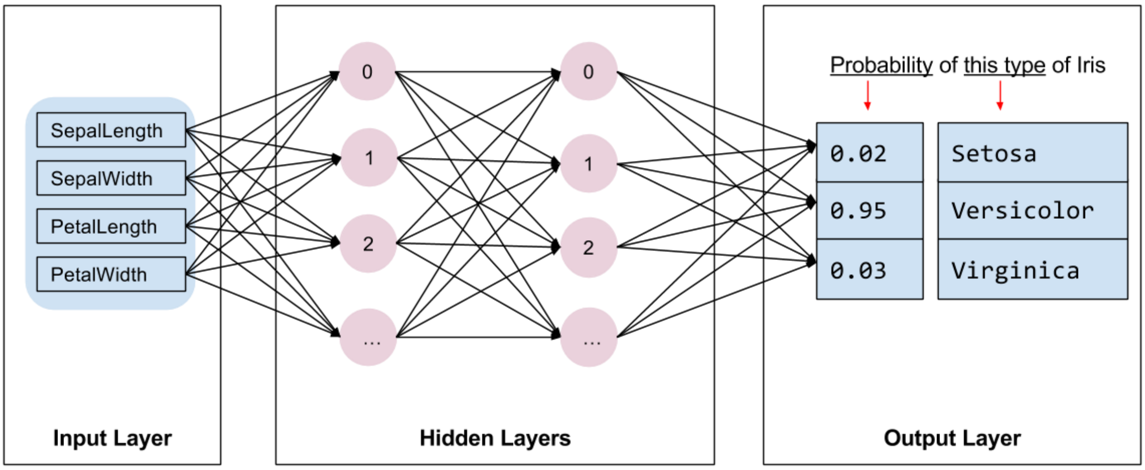
3.1.5 TensorFlow

TensorFlow is a software library for numerical computation through graphs. Nodes represent mathematical operations and the edges represent the data arrays communicated between different nodes. It was originally created during the Google Brain project, and since then it proved to be general enough to be applied in a wide range of cases and domains.

TensorFlow has been applied in a variety of real word cases with success. Some examples are its use on Google for search ranking, computer vision models, automatic generation of email responses, identification of promising drug candidates and optical recognition that enables real-time translation.

A demonstration of the neural networks created by Tensorflow can be seen in Figure 19, with an example of input, hidden layers, and possible outputs.

Figure 19: Tensorflow neural network example.



Source: (TENSORFLOW WEBSITE, 2017)

3.2 OpenAI Baselines

OpenAI Baselines is a collection of high-quality Reinforcement Learning algorithms, already being vastly used in many academic and commercial projects and studies. The full OpenAI Baseline is composed of the following implementations:

- A2C
- ACER
- ACKR
- DDPG
- DQN
- GAIL
- HER
- PPO1
- PPO2
- TRPO

From the available implementations, the Deep Queue Network(DQN) and the Advantage Actor-Critic (A2C) were selected for being the most commonly used in academic projects.

3.2.1 Advantage Actor Critic – A2C

In general, the idea for Reinforcement Learning algorithms is to have an agent that receives the state of an environment and then takes actions while trying to maximize its rewards. The Advantage Actor-Critic implementation fulfills that premise in the following steps.

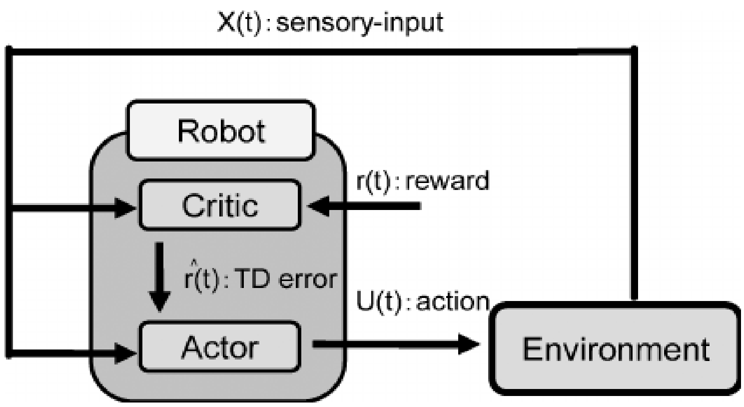
First, it receives the current state of the environment and uses it to generate two outputs. One output is the estimate of rewards he expects to find ahead, which is called ‘state value’ and is meant to be the ‘critic’. The second output is a recommendation of what action to take, called ‘policy’, and is meant to be ‘actor’.

Each step of state-action-reward results in the recording of the state, reward expected, action taken and reward found. After collecting the information of three iterations, the agent adjusts its critic based on the estimates he had before and the results collected, calculating the difference between what he expected, what he received and what lead him to take the action.

The algorithm also makes sure to ensure that the agent won't always choose the safest option but that has a low potential for rewards. That is done by subtracting a value called ‘entropy’ from the loss function, which is responsible for measuring the performance of the actions.

A schematic implementation of the A2C model can be seen on Figure 20, using a robot as an example scenario.

Figure 20: Actor – Critic Model.



Source: (MEDIUM BLOG POST, 2017)

3.2.2 Deep Queue Learning – DQN

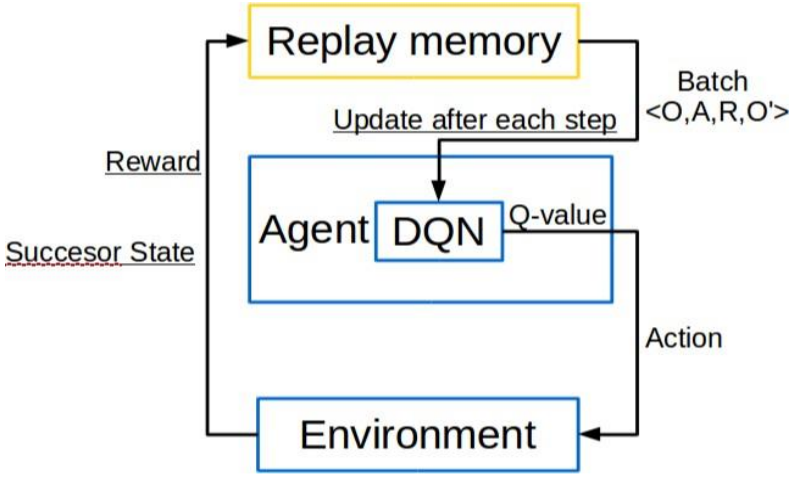
The Deep Queue Learning algorithm is a direct evolution from the Q-Learning algorithm. It was created to solve the lack of generality, which is a weakness present in the Q-Learning implementations. So even though Q-Learning is a very good RL algorithm, that lack of generality makes it unable to estimate the outcome values for states that it has not yet seen.

DQN improves the Q-Learning algorithm by introducing a neural network to estimate the reward values, giving it the ability to store the results already obtained by specific action-state decisions taken. Another improvement is the fact that the DQN implementation uses randomly picked batches of experiences from its pool, to help the network to develop itself with a broad range of experiences.

The third addition to the Q-Learning implementation is the use of a second network during the training process, responsible for reward estimates that are used in the loss function during the training process. By having separate networks for each area, DQN reduces the risk of estimations values spiraling out of control, resulting into a feedback loop of state, reward estimates and actions are taken.

A schematic implementation of the DQN model can be seen in Figure 21.

Figure 21: Deep Queue Network Model.



Source: (DNDDNJS WEBSITE, 2018)

3.3 Communication

The setup necessary for the training requires that multiple separate parts work together, such as the communication between the agent and the game, which allows the agent to send commands to the game and receive game state.

The SC2client-proto is a Machine Learning API created by the SC2 game creator Blizzard, being an interface that offers full control of the game, exposing all the necessary commands and information for agents, bots and replay analyzers to work.

To facilitate the task of creating agents that play SC2, DeepMind created the PySC2, which consists of a set of tools that exposes SC2client-proto API as a Python RL environment package. It provides an interface for developers to create the agents without the need to use direct commands to the game, offering an easy-to-use wrapper for the RL agents to interact with SC2.

The mini-games being used for the agents training consist of SC2 custom maps that were created with specific sets of rules, and can only be run using the SC2 game client. SC2 maps are created using the Blizzard Map creator tool and all the data related to each map is stored in a '.SC2Map' format file.

4 RESULTS

This chapter presents the results and analyses conducted during the final stage of this work.

4.1 Methodology

Training was divided into two groups, defined by the mini-game being played, with each individual group of tests being composed by sessions of the agent playing the mini-game with both RL algorithms and the learning parameters being tweaked.

Each training session was limited to 100 episodes of the mini-game being played by the agent, focusing the attention on the score value obtained by each set of algorithm and parameters during the session.

Each training session data generated was collected as well as the trained model, which allowed the analysis of the score by episode, as well as the use of the pre-trained agent model on side-by-side tests with other agents.

4.2 Agent Setup

The agent has a set of parameters that defines some aspects of its behavior during training, those parameters being independent of the RL algorithm being used. The relevant parameters are the following:

- Discount: reward discount for the agent, can be tweaked to test the reward influence during training;
- Loss Value Weight: how much a loss weight, can be tweaked to test the influence of a loss during the training;
- Entropy: correspond to the spread of actions probabilities, low entropy means one dominant action, while high entropy means multiple actions with similar probability.

Can be tweaked to test how it affects the agent exploration of new actions or strategies.

The agent setup for the training sessions was defined as one setup with default values for all described parameters for both algorithms and one setup with an altered entropy value from 1e-6 to 1e-5 for both algorithms, described in Table 2.

Table 2: Agent Parameters Setup Table

	Default Setup	**Custom Setup**
Discount value	1	1
Loss Weight value	1	1
Entropy value	1e-6	1e-5

Source: AUTHOR (2018)

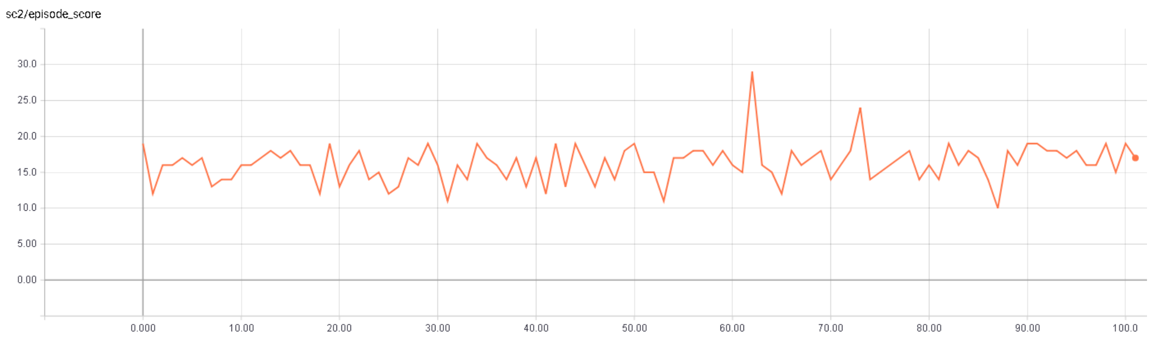
4.3 Results

This section presents all the results collected during the tests, demonstrating the most significant values in each test, resulting graphs and a table of values for comparison.

4.3.1 Collect Mineral results

The first test was conducted using the A2C algorithm with default parameters. The scores obtained by the agent ranged from 10 to 29, resulting in the graph seen in Figure 22.

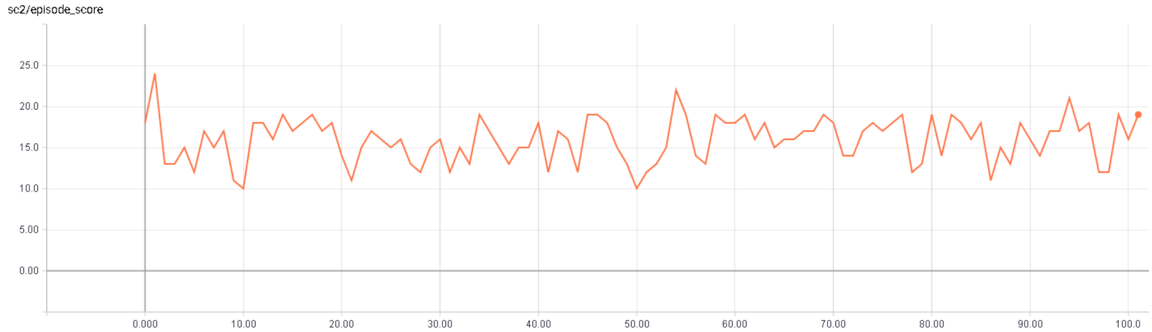
Figure 22: A2C With Default Parameters Results For Collect Minerals



Source: AUTHOR (2018)

The second test was conducted using the DQN algorithm with default parameters. The scores obtained by the agent ranged from 10 to 24, resulting in the graph seen in Figure 23.

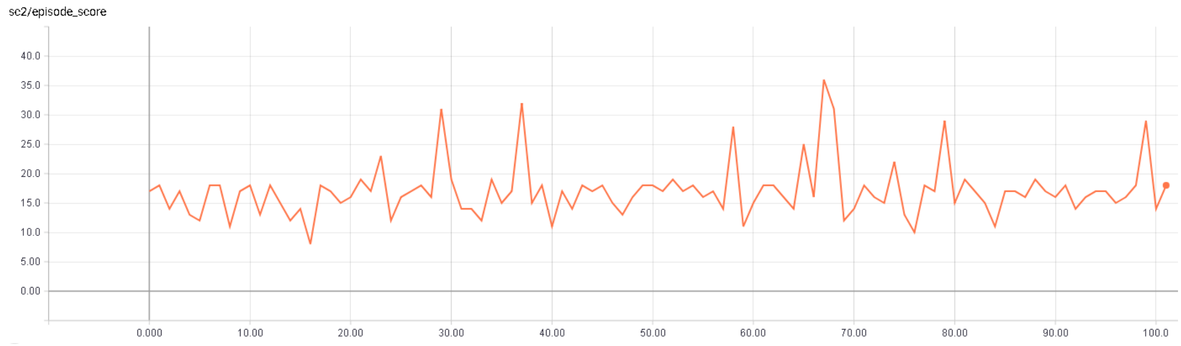
Figure 23: DQN With Default Parameters Results For Collect Minerals



Source: AUTHOR (2018)

The third test conducted was the ‘Collect Minerals’ mini-game using the A2C algorithm with custom parameters. The scores obtained by the agent ranged from 8 to 36, resulting in the graph seen in Figure 24.

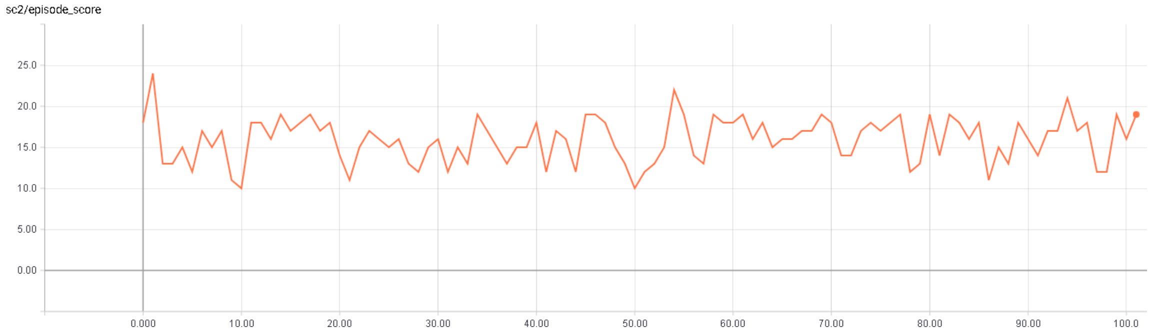
Figure 24: A2C With Custom Parameters Results For Collect Minerals



Source: AUTHOR (2018)

The fourth test conducted was the ‘Collect Minerals’ mini-game using the A2C algorithm with custom value parameters. The scores obtained by the agent ranged from 7 to 33, resulting in the graph seen in Figure 25.

Figure 25: DQN With Custom Parameters Results For Collect Minerals



Source: AUTHOR (2018)

In Table 3 are presented the lowest, highest and average values obtained by each algorithm during the ‘Collect Mineral’ mini-game.

Table 3: Collect Mineral Shards Results Table

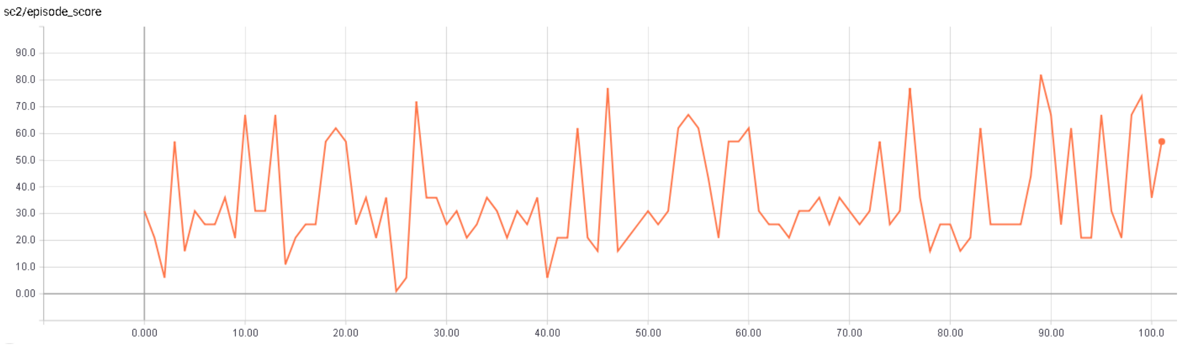
	A2C	**DQN**	**Custom A2C**	**Custom DQN**
Lowest value	10	10	8	7
Highest value	29	24	36	33
Average value	16,18	15,88	17,07	16,65

Source: AUTHOR (2018)

4.3.2 Defeat Zerglings results

The first test was conducted using the A2C algorithm with default parameters. The scores obtained by the agent ranged from 6 to 108, resulting in the graph seen in Figure 26.

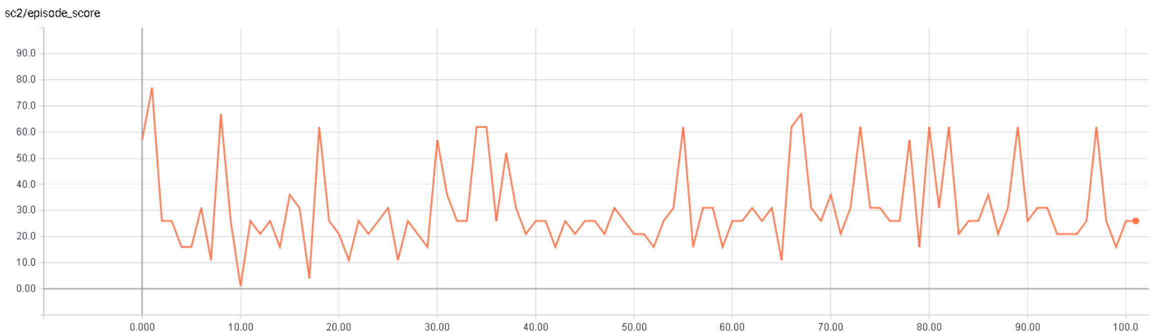
Figure 26: A2C Results For Defeat Zerglings



Source: AUTHOR (2018)

The second test was conducted using the DQN algorithm with default parameters. The scores obtained by the agent ranged from 1 to 77, resulting in the graph seen in Figure 27.

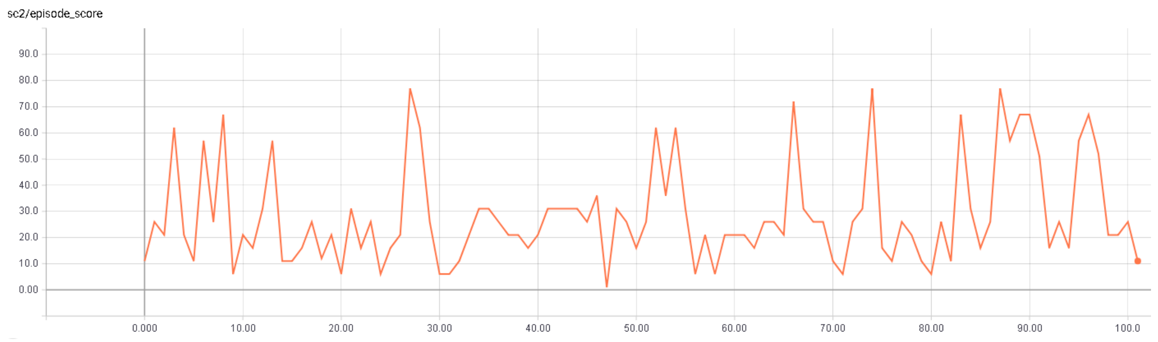
Figure 27: DQN Results For Defeat Zerglings



Source: AUTHOR (2018)

The third test was conducted using the A2C algorithm with custom parameters. The scores obtained by the agent ranged from 1 to 77, resulting in the graph seen in Figure 28.

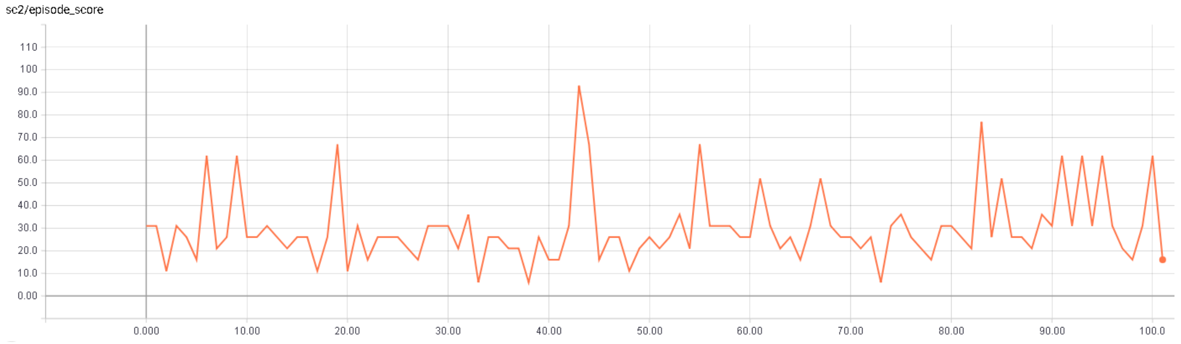
Figure 28: A2C With Custom Parameters Results For Defeat Zerglings



Source: AUTHOR (2018)

The fourth test was conducted using the DQN algorithm with custom parameters. The scores obtained by the agent ranged from 6 to 93, resulting in the graph seen in Figure 29.

Figure 29: DQN With Custom Parameters Results For Defeat Zerglings



Source: AUTHOR (2018)

In Table 4 are presented the lowest, highest and average of values obtained by each algorithm during the ‘Defeat Zerglings’ mini-game

Table 4: Defeat Zerglings Results Table

	A2C	DQN	Custom A2C	Custom DQN
Lowest value	1	1	1	6
Highest value	82	77	77	93
Average value	35,21	30,6	28,59	29,69

Source: AUTHOR (2018)

4.4 Summary

The results from the Collect Minerals mini-game showed that the A2C algorithm fared better than the DQN on both the default and custom parameters tests. The entropy parameter changes seemed to have influenced the resulting values, resulting in a bigger difference range between the lowest and highest scores, which also resulted in higher score average values.

The results from the “Defeat Zerglings’ mini-game showed similar results for the A2C and DQN algorithm, with the A2C faring better with default value parameters while the DQN did better on the custom ones. The entropy parameter changes drove the DQN scores up and the A2C down, resulting in a bigger difference range between lowest and highest scores, which resulted in lower score average values.

Considering the results obtained, the A2C seems to offer the best results when used for Reinforcement Learning agents running Starcraft II mini-games. The parameter changes during the ‘Defeat Zerglings’ that resulted in the DQN doing better at that test, point to the notion that a finer tweak in the parameters can influence the final results, for best or worst.

The test consisted in a total of eight training sessions, four on each mini-game, being two with each algorithm, one with default and one with custom parameters.

Each Collect Minerals training session took close to one hour, while the Defeat Zerglings varied between 30 to 50 minutes because the episodes could end sooner if the agent lost all its units.

The machine used for the tests was a low end desktop configuration, with a AMD Phenom II 955 running on 3.200MHz clock and using 8Gb of DDR3 ram running on 1333Mhz. Taking in consideration the computer specs, it’s reasonable to affirm that the time taken for the test could be considerable lower using a more computationally powerful machine.

5 CONCLUSION

This work has briefly analyzed the current state of two Reinforcement Learning algorithms applied to a complex and modern computer strategy game. Tests were conducted using default and custom agent parameters, offering some preliminary answers about how each algorithm would achieve success on each mini-game and how the parameters might affect them.

While each test session only amounted for a small number of episodes, we confirmed that changes in parameters and algorithm being used on each mini-game do significantly affect the results, even without the agent achieving a convergence point.

The entropy being selected to be the changed parameter showed us that it can have a heavy influence on the results, leaving open the question about how the other parameters would affect the tests.

Related works focus on convergence, comparing algorithms and the time required for each to achieve the convergence point, while this work focused more on fine comparison of algorithm results and parameters on a smaller episode range of episodes.

5.1 Future Work

This work used a reduced amount of episodes for each training session, that decision forces the analyses of the results to look for sharp changes in behavior based on the parameters and algorithm being tested.

Future works that enjoy the benefits of dedicated servers with hardware specific for mathematical calculations could extend the training sessions by a large amount, being able to achieve convergence in results as the agents learn the best strategies for each mini-map

Another interesting task is the study of Reinforcement Learning algorithms that were not tested in this work, followed by the challenge of training agents to play the whole game, which would require it master all of the tasks previous seen individually in the mini-games.

REFERENCES

ADAMS, R. **10 Powerful examples Of Artificial Intelligence In Use Today.** Available at: <https://www.forbes.com/sites/robertadams/2017/01/10/10-powerful-examples-of-artificial-intelligence-in-use-today/>. Accessed on October 23 2017.

ALBRIGHT, D. **10 Examples of Artificial Intelligence.** Available at: <https://www.techemergence.com/everyday-examples-of-ai/>. Accessed on October 22, 2017.

ATTICK, R. **Intelligent Things >> It's all about machine learning.** Available at: <https://www.linkedin.com/pulse/intelligent-things-its-all-machine-learning-roger-attick//>. Accessed on October 22, 2017.

DEEPMIND; BLIZZARD; **StarCraft II: A New Challenge for Reinforcement Learning.** 2017.

DNDDNJS. **Deep Q Networks.** Available at: <https://dnddnjs.gitbooks.io/rl/content/deep_q_networks.html>. Accessed on June 14, 2018.

DOMINGOS, P. **The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World.** Basic Books, 2015.

FELLOW, L; LI, X. **Machine Learning Paradigms for Speech Recognition: An Overview.** IEEE Transactions on Audio, Speech, and Language Processing, 2013**.**

FRAGGELLA, D. **Machine Learning in Robotics.** Available at: <https://www.techemergence.com/machine-learning-in-robotics/>. Accessed on October 24, 2017.

GEITGEY, A. **Machine Learning is Fun Part 6: How to do Speech Recognition with Deep Learning.** Available at: <https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a>. Accessed on October 24, 2017.

IYYER, M; MANJUNATHA,V; BOYD-GRABER, J; DAUME ,H. **Deep Unordered Composition Rivals Syntactic Methods for Text Classification.** University of Maryland, 2015.

KAPLAN, A. **The Epic Evoluition of Video Games.** Lerner Classroom, 2013.

KUBAT, M; HOLTE, R; MATWIN, S. **Machine Learning for the Detection of Oil Spills in Satellite Radar Images.** Kluwer Academic Publishers,1998.

KUMAR, J; HERGER, M; DAM, R. **A Brief History of Games.** Available at: <https://www.interaction-design.org/literature/article/a-brief-history-of-games/>. Accessed on October 25, 2017.

MALISIEWICZ, T. **Deep Learning vs Machine Learning.** Available at: <http://www.computervisionblog.com/2015/03/deep-learning-vs-machine-learning-vs.html>. Accessed on October 24, 2017.

MEDIUM. **Reinforcement Learning using Asynchronous Advantage Actor Critic.** Available at: <https://medium.com/@henrymao/reinforcement-learning-using-asynchronous-advantage-actor-critic-704147f91686>. Accessed on June 14, 2018.

MILLINGTON, I. **Artificial Intelligence for Games.** Morgan Kaufmann Publishers, 2009.

NARULA, G. **Everyday Examples of Artificial Intelligence and Machine Learning.** Available at: <https://www.techemergence.com/everyday-examples-of-ai/>. Accessed on October 22, 2017.

NIELSEN, M. **Using neural nets to recognize handwritten digits. Available at: <http://neuralnetworksanddeeplearning.com/chap1.html>.** Accessed on October 25, 2017.

OPENAI GYM. **A toolkit for developing and comparing reinforcement learning algorithms.** Available at: <https://github.com/openai/gym/>. Accessed on October 25, 2017.

ROCK PAPER SHOTGUN. **Stacraft 2, Wings of Liberty.** Available at: <https://www.rockpapershotgun.com/>. Accessed on June 14, 2018.

RUSSEL, S; NORVIG, P, **Artificial Intelligence – a modern approach**. Prentice-Hall, New Jersey, 1995.

RUUSE, L. **Artificial Intelligence Everything You Want To Know.** Available at: <https://www.scoro.com/blog/artificial-intelligence-everything-you-want-to-know/>. Accessed on October 23, 2017.

SELFRIDGE, M; DICKERSON, D; BIGGS, S. **Cognitive Expert Systems and Machine Learning: Artificial Intelligence Research at the University of Connecticut.** AI Magazine Volume 8 Number 1, 1987**.**

SHAIKH, F. **Simple Beginner's guide to Reinforcement Learning & its implementation.** Available at: <https://www.analyticsvidhya.com/blog/2017/01/introduction-to-reinforcement-learning-implementation/>**.** Accessed on October 25, 2017.

SONKA, M; HLAVAC, V; BOYLE, R. **Image Processing, Analysis, and Machine Vision.** Cengage Learning, 2014.

SUTSKEVER, I; VINYALS, O; LE, Q. **Sequence to Sequence Learning with Neural Networks.** Cornell University Library, 2014.