

# Enhancing Hearthstone deck building with a Generative Adversarial Network (GAN)

Callum ROBERTS

April 3, 2021

## Abstract

## Acknowledgements

I would like to thank, in particular, my project supervisor Mark Bartlett, whose wisdom proved invaluable to the completion of this project. This thanks extends to the rest of the RGU staff who went beyond expectations in their support during these tough times.

Thanks are also owed to the Hearthpwn team who allowed me to collect their data and to the HearthSim team for their simulator to run the tests.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Overview . . . . .	3
1.2	Motivation . . . . .	3
1.3	Aims and Objective . . . . .	3
1.4	Key Findings . . . . .	4
1.5	Structure . . . . .	4
<b>2</b>	<b>Literature Review</b>	<b>5</b>
2.1	Background . . . . .	5
2.1.1	Collectible Card Games . . . . .	5
2.1.2	Hearthstone . . . . .	6
2.1.3	Deck Building . . . . .	7
2.2	Project Introduction . . . . .	10
2.2.1	Related Works . . . . .	10
2.2.2	Research Questions . . . . .	11
2.2.3	Assistance Systems . . . . .	12
2.3	Genetic Algorithm . . . . .	13
2.4	Machine Learning . . . . .	14
2.4.1	Artificial Neural Network . . . . .	14
2.4.2	Generative Adversarial Networks . . . . .	16
2.5	Hybrids . . . . .	17
2.6	Conclusion & Final Thoughts . . . . .	17
<b>3</b>	<b>Methodology</b>	<b>18</b>
3.1	Overview . . . . .	18
3.2	Functional Requirements . . . . .	18
3.3	Non-Functional . . . . .	19
3.4	Approach . . . . .	20
3.5	Libraries . . . . .	20
3.5.1	Web Scaper and API . . . . .	20
3.5.2	GAN . . . . .	21
3.5.3	Evaluation . . . . .	21

# 1. Introduction

## 1.1 Overview

Bringing Artificial Intelligence to Hearthstone: Heroes of Warcraft is not a new concept, it has been subject to many studies in the field, where it be playing the game, suggesting moves to the player or building decks. Focusing on the field of deck building, it is a concept that exists across multiple trading card games such as Magic: The Gathering, Pokemon, Yu-Gi-Oh! and of course Hearthstone. Papers have studied these games more or less, but all seem to gravitate around similar deck building techniques, details of which will be expanded later on. These techniques seemingly been saturated, newer studies apply small scale changes to already defined methods [19]. Exploration of different techniques has been touched on but they seem to be outliers and of limited number [34], experimenting with other algorithms could show better, more interesting results instead of assuming that one saturated method is the best because of its popular usage.

## 1.2 Motivation

Using a Artificial Intelligence algorithm that has not been utilised nor researched in the field of deck building could provide insight into the Generative Adversarial Network (GAN) applications in other fields of research, as it is mostly limited to image generation. Along with this discovering another method for deck building that could bear fruit to similar or improved results, opening the way for more techniques and deeper research into the use of recorded data for deck generation.

## 1.3 Aims and Objective

The aim of this project is to create a deck building artificial intelligence using a Generative Adversarial Network (GAN) and testing the viability of it, this will be achieved by completing the following objectives:

- Collecting user deck data from reliable sources

- Cleansing of user deck data
- Converting user deck data to vectors and back to human readable after training
- Implement GAN for single dimensional vector generation
- GAN hyper-parameter and layer optimization
- Simulating results against user decks
- Performing evaluations on resulting decks

## 1.4 Key Findings

This project has exhibited that it is possible to use Generative Adversarial Networks to generate decks for Hearthstone, a technique that seemingly no one has attempted to use, the AI demonstrates similarities to user created decks such as card types, type percentage, card duplicates, synergies and card spread. The decks tested have an average of 55% win rate, varying from class to class, match ups against other classes greatly influenced the outcome of a match. Although the win rate is heavily influenced by the AI playing it, considering that the simulator game playing AI will not play as well as a human player. The training process is much faster than anticipated (around 10 minutes), however the testing was much longer at around 1 - 2 hours depending of the number of decks created.

## 1.5 Structure

The structure of the report is as follows, first a literature review which presents an overview of Hearthstone, some of the important game mechanics and existing AI that have been used. Then a discussion of the requirements of the project and the methodology to implement them. Following this is a discussion of the potential legal and ethical issues with the project and what has been done to overcome them. The implementation of the project, a deep dive into how the project was completed. A testing and evaluation section describing the strategy and the results of the generated decks. Finally, a conclusion which summarises the project is given.

## 2. Literature Review

The purpose of this literature review is to define the technologies used in the field of artificial intelligence for building a deck in Hearthstone. To introduce, and compare previous works to determine their strengths and weaknesses. A review of the literature is valuable in understanding important aspects of a research area [1]. The structure of this literature review is as follows: the initial section will detail the background of the project, explaining the fundamentals of Hearthstone and deck building. Following that will be an introduction to the project, motivations, and research questions. Finally, we will have the core technologies used for similar projects.

### 2.1 Background

For the benefit of the reader, this section will introduce the basics of Hearthstone. It will emphasize the deck building aspect of the game including practices used by players.

#### 2.1.1 Collectible Card Games

Collectible Card Games (CCG)<sup>1</sup> are a sub-genre of card games introduced in 1993 by Magic: The Gathering<sup>2</sup>. They require players to make a custom decks to play, they mix trading cards with strategy and deck building features. CCGs are usually defined as a turn-based game, where each player acquires their own collection of cards through the purchasing of "starter decks" for beginners or "booster packs" containing a small number of random cards from a *pool* of cards usually referred to as an expansion. The aim is to build an efficient deck that can account for the inconsistency that comes from the nature of card games, to predict and play around your opponent's actions to ultimately beat them. Some CCGs can prove to be lucrative for players as cards have a value intrinsic to their rarity and demand<sup>3</sup>, this makes building the perfect deck rather difficult and usually costly.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Collectible\\_card\\_game](https://en.wikipedia.org/wiki/Collectible_card_game)

<sup>2</sup>[https://en.wikipedia.org/wiki/Magic:\\_The\\_Gathering](https://en.wikipedia.org/wiki/Magic:_The_Gathering)

<sup>3</sup><https://www.cardmarket.com/en/Magic/>

### 2.1.2 Hearthstone

Hearthstone: Heroes of Warcraft is a CCG developed by Blizzard Entertainment in 2013 [2], but with the twist of it being entirely digital, there is no physical version of the game. This choice unlocks potential for gameplay features that could not be implemented, in exchange for the tradability of cards.



Figure 2.1: Example of a Legendary Hearthstone card<sup>4</sup>

Two players face off wielding each a deck of their own making. Decks consist of exactly 30 cards. Players then take it in turns to play their cards, the objective being to reduce the other player's health to zero. On each player's turns that player draws a card and gains a "Mana Crystal" up to a maximum of 10 (crystals are refreshed every turn), these crystals are expended to cast a card from the player's hand. Before a match each player chooses to embody a class (such as Mage, Warrior, Rogue, Druid, etc. . . ), each class has specific cards only they can add into their deck, these are adequately named "Class Cards", these are accompanied by "Neutral Cards" that any class can use. Classes also have access to an ability unique to them called a "hero power".

Each card in the game has a "mana" cost which shows how many mana crystals are needed to cast that card. They also have a card type, rarity, and an effect. In a deck, players can put duplicates of the same card (up to 2) except for "Legendary Cards" (figure 2.1) that are limited to a single copy due to their powerful effects. Players have a "Collection", where the cards they own are stored, to get new cards players can buy card packs with gold, the in-game currency of Hearthstone. Gold is earned slowly through quests, winning, and events, however this process can be sped up through the purchase of gold with real-life currency. Players can also choose to "Disenchant" their duplicate cards to gain another in-game currency called "Dust" which can be used to create a

<sup>4</sup><https://www.pinterest.fr/pin/573716440004576557/>

<sup>5</sup><https://bothgunsblazingblog.wordpress.com/2014/06/22/hearthstone-analysis-and-deconstruction/>





Figure 2.2: Example of a Hearthstone board<sup>5</sup>

card of the players choosing<sup>6</sup>.

### 2.1.3 Deck Building

In the world of CCGs, there is a long-standing debate on how to measure the skill of a player. Although card games involve luck and circumstance, it is believed that there is a degree of strategy in the building and execution of decks whether it is just a slight increase in win probability or a fundamental to winning [3]. However, the debate stems from which is the most important, the building aspect or the execution aspect of CCGs. [4]

#### 2.1.3.1 Metagame

Hearthstone is a game with lots of complex systems that are influenced by many factors, mainly due to a large number of cards and different playable heroes. In a game where there are lots of variables, players try to rank cards, heroes, and combinations to increase their chances to win. This phenomenon creates decks from a "pool" of top-rated cards, leaving out the mediocre, forcing players to use these top-rated decks in order to have a better chance of winning or be put at a disadvantage. The result is what is called the "Metagame" or *meta* for short<sup>7</sup>. Blizzard release updates to the game frequently through "expansions"<sup>8</sup> which add a variety of new cards to keep the game fresh. Shifts

<sup>6</sup><https://hearthstone.gamepedia.com/Crafting>

<sup>7</sup><https://www.hearthstonetopdecks.com/hearthstones-best-standard-ladder-decks/>

<sup>8</sup><https://hearthstone.gamepedia.com/Expansion>

in the meta occur when these expansions are added and players experiment to find better combinations over time. However better cards may not be added, and changes in the meta may not occur, this dissuades players from continuing or returning to play knowing that they have already experienced all that they can. To avoid that Hearthstone has implemented two-game modes<sup>9</sup>, one in which only cards added over the past two years are available, and another mode that allows all cards. Whilst this method has helped, it still does not put a stop to the possibility of a stale meta. Researchers have done studies on how to evolve the meta through AI means, by *balancing* powerful cards (Fernando et al.) [5]. Balancing a card means to adjust the power of said card to make it more or less viable in the current game environment. Fernando et al. discussed the idea that around 50% of Hearthstone’s meta is derived from match-ups which is the win probability two decks have against each other, a favorable match-up being the one with the highest win probability or known in the community as *win rate*.

### 2.1.3.2 Mana Curve

Theorycrafting is a term used widely in many video games, it designates a mathematical analysis of a game’s core mechanics to attempt to discover new strategies or combinations that could rival the current ones. Hearthstone is one such game, a large portion of the player base enjoys theorycrafting new decks that may *break* the meta, as in cause a fundamental shift of the current metagame. These players rely on fundamentals or schematics that are used as a guideline in building a new deck. The *mana curve*<sup>10</sup> is one such fundamental, it exists in all decks built in the game. Every card in Hearthstone has a cost, this cost determines the power of the card, a low-cost card will be weaker than a higher cost card since it would cost fewer resources to cast. This *mana curve* is a histogram of each card plotted by cost, it allows players to visualize how expensive in resources their deck is and to determine the deck’s archetype.

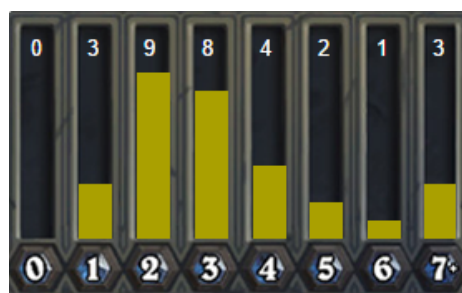


Figure 2.3: Example of a Mana Curve<sup>11</sup>

<sup>9</sup>[https://hearthstone.gamepedia.com/Game\\_format](https://hearthstone.gamepedia.com/Game_format)

<sup>10</sup>[http://hearthstone.gamepedia.com/Mana\\_curve](http://hearthstone.gamepedia.com/Mana_curve)

<sup>11</sup><https://hearthstone.judgehype.com/deck-mage-tempo-ladder-legendaire-tgt-gvg/>

### 2.1.3.3 Archetype

The word archetype is derived from the Greek word *archétypon* which means "beginning, origin", applied in the psychology field to categorize complex human behaviour called "Jungian Archetypes" [6]. This term was transposed into the deck building field of CCGs, a deck's archetype is meant to categorize and describe the behaviour of the deck from a high-level perspective, forgoing the need to play the deck to learn its strategy. In Hearthstone, most decks can be categorized by three main archetypes [7]:

- *Aggro* decks are the aggressive decks meant to defeat an opponent as quickly as possible, as a consequence the mana curve of such decks is focused towards the cheaper side of the histogram. Their power comes in the early turns, but they quickly become weaker to the other archetypes as the turns go on.
- *Control* decks are meant to control the state of the board through the use of expensive cards, they tend to generate a lot of cards and have a wide range of card choices. The mana curve of such an archetype is towards the expensive end of the histogram. They tend to have a few cards to play early in the game but have a multitude of win conditions in later turns.
- *Mid-range* decks are situated in the middle of the two other archetypes, focusing mainly on the mid-game, their win conditions are stronger than the aggro decks but weaker than the control decks. Their mana curve peaks in the middle of the histogram.

Hearthstone also has other archetypes that cover a shorter scale, they are usually introduced in the newer expansions and rotate out of the normal game after a couple of years [8]. For example, a highlander archetype is a deck with 30 unique cards. Archetypes are formed around specific cards with win conditions<sup>12</sup>, meaning that they have the power to win the game. So in the example given in a highlander deck, there would be a card that has an effect that triggers from having only unique copies in the deck.

### 2.1.3.4 Resource Cost

In CCGs, cards have a certain cost to use, in Hearthstone that cost is mana which regenerates every turn. The cost of a card is determined by the power of said card, if it has a powerful effect, has decent attack and defence values, or even both. This cost will determine how late into the game a card can be played. However, a card that costs a lot can be considered weak and a low-cost card can be considered strong. The power of a card is determined through the resource cost, an invisible value that is hard to calculate and a subject of study [9]. Although the Zuin et al. study was used to predict the cost of a card in Magic: The Gathering, the resource calculation is still present in Hearthstone. It poses

---

<sup>12</sup><https://playhearthstone.com/en-us/news/21363038>

a good solution to the balancing of the metagame and would be adaptable to Hearthstone. A card is considered efficient if the theoretical resource cost is higher than the current mana cost, and would be inefficient if the resource cost were to be lower than the mana cost. The resource cost of a card is something that may need to be considered when developing an AI for building decks. Stiegler et al. [10] applied a similar theory to design a deck-building AI based on a utility system that classified cards based on resource cost-effectiveness, mana curve, and synergies.

## 2.2 Project Introduction

The rising popularity of Hearthstone has attracted a lot of new players reaching over 100 million accounts in 2018 [11], however, due to the nature of collecting cards in the game some will not have the required cards to build the most popular decks. The game is free, anyone can download it, however, a lot of the content is locked behind a paywall<sup>13</sup>. Whilst it is possible to earn cards by earning "gold", it becomes a time-consuming ordeal that requires a lot of spare time to invest. With new expansions being added regularly, the game seems to become a never-ending grind, unless you decide to pay real money to acquire currency. This is where the term "pay-to-win" is used to describe Hearthstone [12], meaning that to get the most enjoyment and the highest chance to win, the player must spend money or be disadvantaged. The goal of this honours project is to create an AI that builds decks from a collection of cards, incomplete or otherwise in order to improve the game experience for players that are unable or not willing to pay. For players that do own a large collection of cards, it can also provide fresh new decks to play that differ from the more popular ones.

### 2.2.1 Related Works

Video games are the ideal tool for the training of Artificial Intelligence. The virtual space that a game provides is a realistic environment with a limited amount of information available [13] allowing control and knowledge over the behaviour of the AI. Hearthstone is a game that provides a platform for a wide variety of AI that differs from AI-benchmark games such as Chess or Go. Hoover et al. [14] classifies Hearthstone AI into specialized categories:

- Game Playing AI, rather self-explanatory, this form of AI is designed to play the game. Generally, tree search algorithms are used, Monte Carlo Tree Search (MCTS) in particular. However, this method is rather ineffective in Hearthstone due to the amount of hidden information and limited visibility of the AI. Developers of Hearthstone simulators, such as *MetaStone*<sup>14</sup> tend to use a greedy approach to compensate [15]. Some researchers attempt to use variations of MCTS and heuristics to work around the limited information [16][17][18].

<sup>13</sup><https://en.wikipedia.org/wiki/Paywall>

<sup>14</sup><http://www.demilich.net/>

- Developer Assisting AI, this AI help with certain issues that developers could have. Since Hearthstone has hundreds of cards, it is challenging to design cards with new flavour that are not identical to previously printed cards. Could there be a way to generate inspiration? Woolf "minimaxir" Max<sup>15</sup> created an API that generates Magic: The Gathering cards<sup>16</sup> using a transformer language model<sup>17</sup> for such a purpose. Another possible use is for balancing the game, since maintaining game balance when creating additional cards may create unfair combinations, or render some cards useless[5].
- Deck Building AI, these create decks for the player or another AI to use, most commonly created with Evolutionary Algorithms[19]. It has the inherent advantage of being usable in conjunction with other AI. Such combinations help ascertain potential balance issues without human bias involved. Since this is the main topic of this paper, a further in-depth explanation will be provided in the body.

Whilst all these AI are used in the context of Hearthstone, they are utilized for different aspects of the game, therefore, proving that Hearthstone is a platform with a constant influx of AI challenges to be met, a prime example is the additional *battlegrounds* gamemode<sup>18</sup>, a variation of the game where the creatures attack on their own automatically, then completing your board as you progress between rounds. This alteration of the way the game is played will surely become the subject of a paper in the future.

### 2.2.2 Research Questions

Research Questions are essential to any methodical research, it is the first step in any project and fundamental to any successful project. Kowalczyk[20] described Research Questions as a metaphor for a house: "Your data collection forms the walls and your hypothesis that guides your data collection is the foundation. So, what is the research question? It is the ground beneath the foundation. It is what everything in a research project is built on. Without a question, you can't have a hypothesis. Without the hypothesis, you won't know how to study what you're interested in." The research questions in this literature review are defined as:

- **RQ1:** What are the current best deck-building techniques in Hearthstone?
- **RQ2:** What are the strengths and weaknesses of the different techniques?
- **RQ3:** With our findings, what techniques can be applied to optimize the deck-building problem in Hearthstone?

<sup>15</sup><https://minimaxir.com/apps/gpt2-mtg/>

<sup>16</sup>For example cards: [https://github.com/minimaxir/mtg-gpt-2-cloud-run/tree/master/generated\\_card\\_dumps](https://github.com/minimaxir/mtg-gpt-2-cloud-run/tree/master/generated_card_dumps)

<sup>17</sup><https://openai.com/blog/tags/gpt-2/>

<sup>18</sup><https://hearthstone.gamepedia.com/Battlegrounds>

### 2.2.3 Assistance Systems

Despite AI being widely used in Hearthstone for research purposes, it is against Blizzard's Terms of Service (ToS) to use game-playing AI in Hearthstone (Section 1.C.II)<sup>19</sup>. However, a surge of Hearthstone deck tracking software<sup>20</sup> is being used by players without being banned. So how do players use this kind of software without violating ToS? It was revealed that turning on debug logs would provide enough information for these systems without breaking ToS [21] which birthed a whole sub-genre of AI coined as "Assistance Systems" designed to be used to assist the player without it being considered cheating. This brought on the creation of deck trackers, which mentioned above track which cards each player has used and tracks statistics. Whilst deck trackers are not AI since they just read logs, Bursztein [22] used this system to create an AI that predicted what cards the opponent would play in future turns, and used this predictor AI to climb to *legend* rank (the highest rank in competitive mode<sup>21</sup>). While it was not against ToS to use it, when they presented the tool, Blizzard reached out to them and asked them not to release the code as it was *game breaking*. The effectiveness of the tool was however limited to later turns, the accuracy is much lower (going as low as 50%) in first turns and becomes more accurate each turn. Some of the most crucial turns for some archetypes are in those early turns, so this AI would only maximise effectiveness for the *Control* Archetype (2.1.3.3).

Other assistance algorithms include Hearthstones Arena game mode<sup>22</sup>, a mode in which the player drafts a deck one card at a time by selecting 1 of 3 possible choices from a pool of cards, using Apriori algorithms [23] such as HearthArena<sup>23</sup> to make suggestions based on data from a diverse range of high-quality decks created by player and/or deck building algorithms[24]. However, this sort of algorithm would only be useful in an environment where the player cannot select their cards.

The use of assistance systems is interesting but still requires the interactivity of a third party to function. The advantage to assistance systems is the ethical and legal implementation from it. The disadvantages such as the low accuracy rate of the prediction tool in earlier stages of the game, or the limited usability of the Apriori algorithm makes it difficult to be used in a standard deck-building format.

---

<sup>19</sup><https://www.blizzard.com/en-us/legal/fba4d00f-c7e4-4883-b8b9-1b4500a402ea/blizzard-end-user-license-agreement>

<sup>20</sup>Some software examples: <https://hsreplay.net/downloads/?hl=en>  
<https://go.overwolf.com/firestone-app/>  
<http://hearthstonetracker.com/>  
<https://trackobot.com/>

<sup>21</sup><https://hearthstone.gamepedia.com/Ranked>

<sup>22</sup><https://hearthstone.gamepedia.com/Arena>

<sup>23</sup><https://www.heartharena.com/>

## 2.3 Genetic Algorithm

Developed AI algorithms often draw inspiration from biology[25], Genetic Algorithms (GA) is an example of this. GAs are a subset of Evolutionary Algorithms (EA) that base their training process the same way nature does, biological evolution through natural selection (Figure 2.5). A population of solutions each with a set of properties (chromosomes in nature) that are mutable is randomly generated. Each iteration or *generation* the algorithm selects the fittest individuals of the population using a fitness function, then the most fit are used to form the next generation. Some mutations of properties may occur in some of the population. The algorithm ends when either the correct fitness level is achieved or when the number of set generations is reached[26]. GAs are stochastic in nature, meaning that a single iteration would not be sufficient to provide significant statistical results [27]. The process is reminiscent of Charles Darwin’s theory of evolution<sup>24</sup> and proves to be effective in optimization problems[25]. This algorithm is the most frequently used for deck building

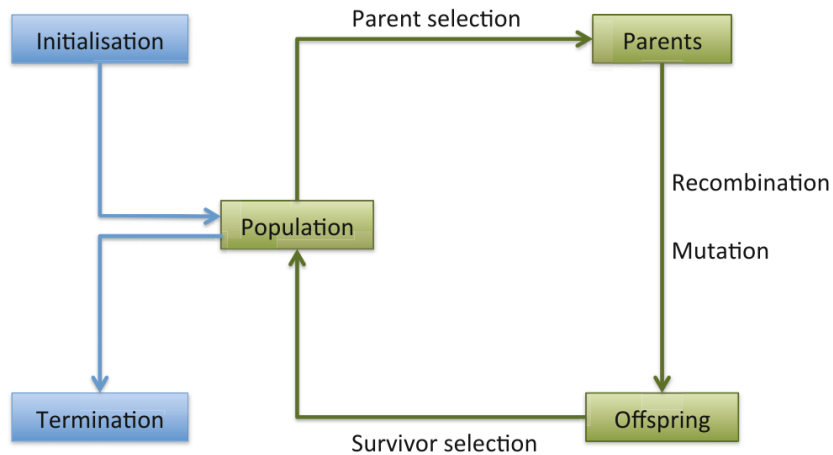


Figure 2.4: Evolutionary Algorithm Flowchart [25]

problems [28][29][30], although they are optimized in different ways. In GAs, there exists a fitness function that determines the fitness score of an individual that is used to create the next generation, and there is the mutation function which will randomly mutate some individuals (it may or may not improve the fitness of said individual).

Bjørke and Fludal [28] used a genetic algorithm to construct decks for Magic: The Gathering based on a certain pool of cards. Instead of using a fitness function that would calculate the score of a deck, they pitted each deck in that gener-

<sup>24</sup>[https://en.wikipedia.org/wiki/Natural\\_selection](https://en.wikipedia.org/wiki/Natural_selection)

ation against each other in a tournament format using a Magic: The Gathering game simulator, each deck had the same number of games to play and they would select the fittest decks based of their win rate in the tournament. While theoretically, the idea is sound, the time that it took was substantial for an unremarkable win rate (less than 60%). With 50 matches per deck over 350 generations, it took 43 hours to execute. Due to the time it takes, it would be unusable for players as it would simply take too long.

Garcia-Sanchez et al. had a similar approach using lexicographical fitness with a Hearthstone game simulator called *MetaStone*<sup>25</sup>, separating the fitness evaluation into three parts: one part counted the number of victories of in 16 games, another part which determined the deck correctness (no more than 2 duplicates, only 1 legendary, etc...), and the last part was applying standard deviation to the number of victories, it being optimal if the deck won against every opponent [29]. The results were achieved faster much faster and were of a better standard than Bjørke and Fludal’s work[28], however, the results were not as high as they could have been, mainly due to the fitness function using *MetaStone*’s greedy AI to play. The decisions made by the AI would be greedy and different from that of a human player. The mutation function could also have been touched on, allowing the mutation to make smarter decisions about which cards to mutate.

Garcia-Sanchez et al. tackled the problem once again and touched on the mutation function [30], they developed a *smart mutation* function that would replace a card in a deck with another of a similar cost (roughly  $\pm 1$ ). The results with the smart mutation were overall better than without it. This may solve one of the weaknesses of their previous work[29] but still uses the same greedy simulator heuristic.

## 2.4 Machine Learning

Machine Learning is a subset of Artificial Intelligence that constructs systems that can learn and improve without the need to be explicitly programmed. Burkov described it as "a subfield of computer science that is concerned with building algorithms which, to be useful, rely on a collection of examples of some phenomenon." [31]

This section of the review will present Machine Learning techniques that have been used or could possibly be used for deck building.

### 2.4.1 Artificial Neural Network

Artificial Neural Networks (ANN) also known as Neural Networks (NN) for short is an algorithm that was inspired by the biological neural networks seen in brains. They are a group of interconnected nodes or *neurons* typically organised into multiple layers, an input layer, an output and  $n$  amount of hidden layers. Through many loops known as *epochs*, the NN trains itself using data

---

<sup>25</sup><https://github.com/demilich1/metastone>



from datasets to ultimately make a prediction based on given data as an input. Each node is weighted, and these weights are updated throughout the training process increasing the weight of positive output and decreasing on a negative one, allowing the system to make more accurate predictions [32]

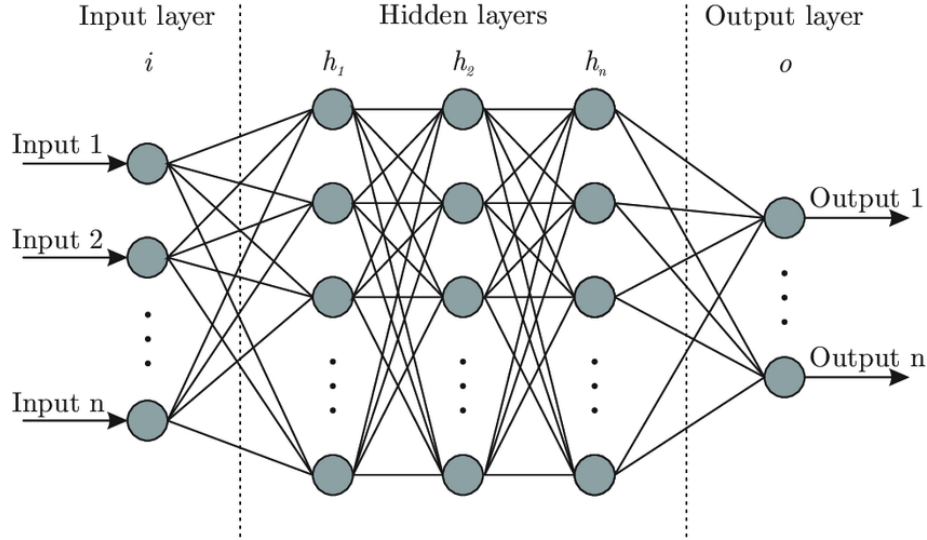


Figure 2.5: Artificial Neural Network architecture [33]

The approach to developing a deck-building AI is done differently than a GA. Ward et al. created a NN that emulated the choices a human would make when drafting a deck from a pool of cards for Magic: The Gathering [34]. The idea was to select the cards that were chosen by a human in the dataset (target variable). The resulting accuracy on the test set was 65.7%. The main drawback to this technique is that it requires the data to be clean, since the dataset used was a lot of decks drafted by human players, the data within may be suboptimal or purposefully tampered with, which would cause the AI to build weaker decks.

Jakubik attempted a different method by using a NN to predict the win rate of a Hearthstone deck learned by using the results of observed matches [35]. This was a proposed solution to the AAIA'18 data mining challenge and came second. However, Jakubik's solution was subject to over-fitting, which Hieu Vu et al. tackled during the same challenge, which brought them the winning solution to the AAIA'18 data mining challenge[36]. They tackled over-fitting by doubling the number of input features (or nodes) to account for the opposing player, they added an extra hidden layer and more test cases to compensate. Jakubik [35] and Hieu Vu [36] rely on data of games that have already been played meaning that this method is not suitable for the construction of the decks rather the prediction of the win rate based on games that have been played. Ward et al. paper is the better technique of the three papers to build good decks, however,

it requires clean and uncorrupted data.

### 2.4.2 Generative Adversarial Networks

Generative Adversarial Networks (GAN) is an application of generative modeling originally described in 2014 by Goodfellow[37], which is an unsupervised learning task that automatically discovers patterns in the input data that the model can then use to output new examples that could have been drawn from the original data. A GAN uses a generative model to generate a similar output and a discriminator model uses supervised learning techniques to determine the probability of the generative output to be part of the training data[37]. The goal of the generative model or *generator* is to create an output that can trick the discriminator into thinking it was part of the dataset. Goodfellow et al. use a fake money analogy: "We can think of the generator as being like a counterfeiter, trying to make fake money, and the discriminator as being like police, trying to allow legitimate money and catch counterfeit money"[38]. The GAN trains itself like this until the generator can create something that tricks the discriminator. Figure 1.6 shows the architecture of a GAN

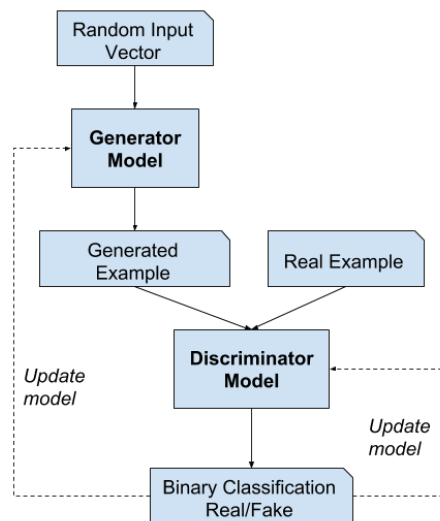


Figure 2.6: Generative Adversarial Network architecture<sup>26</sup>

As of writing this, there have not been any applications of a GAN for building decks. However, Rodriguez Torrado et al. combined a GAN with a procedural content generator to create different playable levels that were randomly generated from the generative model[39]. The GAN struggled to make playable

<sup>26</sup><https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/>

and unique levels when provided a small amount of data. With this paper, it is theoretically possible to apply a GAN in a deck-building context but it will require a large amount of data to provide malleable results.

## 2.5 Hybrids

In AI, aspects of a problem to be solved can be optimised through the use of multiple AI. This section of the review discusses papers that have used a combination of machine learning and evolutionary algorithms. The reason why GAs are combined with Machine Learning techniques is that GAs do not require lots of data to function, this allows for uses such as parameter optimization, Sehgal et al.[40] uses GAs to speed up the learning process of their Deep Reinforcement Learning (RL) agent and provided the maximum success rate faster. Such et al. [41] used it for similar work, and stated that it was a competitive alternative to popular RL training techniques. It has also been observed that in the chemical field that the GA could outperform generative models in a discriminator neural network [42], exploring the possibility of an application in GANs. The use of genetic algorithms in hybrid neural networks is mainly for achieving results faster [40][41], however in highly parametrized systems the GA tends to struggle to find the optimal combination. [43]

## 2.6 Conclusion & Final Thoughts

This literature review has covered existing deck building solutions from Hearthstone and Magic: The Gathering, including evolutionary algorithms, neural networks, and various assistance systems. Potential solutions such as hybrid algorithms and Generative Adversarial Algorithms have also been discussed. Based on this literature review, it has been determined that evolutionary algorithms are the most popular method of tackling the problem therefore the most saturated of new solutions to explore. Machine Learning is a more obscure usage, focusing mainly on win rate prediction and card ranking. Hybrid algorithms and GANs have not yet been used in a deck-building context. In the context of GAs and NNs, they attempt to solve the deck building problem from different perspectives, the GA attempt to build a deck directly and test it in a simulated environment to determine the power of the deck in real-time, whereas the NN focuses on previous data that it uses to predict the power of the deck. They achieve the same goal, however, GAs are more direct and provided concrete results whereas the NN focuses on an indirect prediction from previous use cases. Moving forward this paper will explore solutions within those methods.

## 3. Methodology

### 3.1 Overview

The project implemented will be written in Python, in the form of three python notebooks, each with a specific task. The first notebook collects card data from an API and creates a card dataset, the second notebook scrapes deck data to create datasets for each class, and finally the last notebook will house the AI and the simulator testing. A user simply needs to run the notebooks to get their results, an input is required when deciding what class the user wants the deck to be. With the second notebook, additional datasets can be created and used in the place of the old ones for training the AI.

This project outputs JSON files for the card and deck data, one file for the card data and ten files for the deck data (one for each class). When executing the AI, the user is presented with a certain number of decks, the number varies on how well the GAN trained, these decks are tested and the user is shown graphs and charts of deck composition, overall winrate and winrate per class. They can see the cards within the deck to use at their own leisure.

### 3.2 Functional Requirements

- a. AI **MUST** output a legal deck following the rules of Hearthstone. Meaning that a deck should be composed of 30 cards, with no more than 2 duplicates of any card (excluding legendaries, which can only have a single copy).
- b. AI **MUST** output a deck following the rules of the "Standard" game mode. Meaning that the AI can only use the classic card set and the latest expansions from the past 2 years.
- c. AI **MUST** use a generative adversarial network in order to generate a deck. The network will include a generator to create a similar and a discriminator to determine if the deck is part of the dataset.
- d. AI **MUST** use a dataset of user created decks from the latest expansions in standard format (Rise of Shadows, Saviors of Uldum, Descent of Dragons).

ons, Ashes of Outland, Scholomance Academy, Madness at the Darkmoon Faire) collected from Hearthpwn.

- e. AI created decks **MUST** be tested in a simulator to output a predicted win-rate and determine the playability of the deck versus user created decks. A total of 16 games should be played for optimal results. This will allow for the completion of requirement "f".
- f. The generated deck **SHOULD** have a calculated win-rate of over 50%. Ideally around 60%, so that the AI can make worthwhile decks. Meaning that the created decks should win at least half of the time, and in an ideal scenario they would win more often than lose.
- g. AI **SHOULD** output a dataset of decks it generated following the same format as the training data, as to log the output and determine a possible bias.
- h. AI **SHOULD** be able to generate a deck from a limited collection of cards, so it can be useful to players without access to all cards in Hearthstone.
- i. AI **COULD** be used in the wild format, making use of all expansions to create competitive decks from a larger pool of cards.
- j. An interface **COULD** be created to accommodate the AI and assist in feature implementation and accessibility of users.

### 3.3 Non-Functional

- a. **MUST** have written consent from Hearthpwn to scrape data for dataset
- b. Data collected **MUST** conform to GDPR regulations and be anonymous, since the identities of decks created by users is not needed.
- c. **MUST** use Python as the programming languages of choice for the access to expansive libraries for data manipulation and AI creation.
- d. **MUST** use Jupyter notebook as the software for coding the AI. It provides a platform for Python, and the cells are useful for making small changes without having to re-execute the whole program.
- e. Dataset size **MUST** not be more than 100,000 rows as to decrease time to train and prevent overfitting.
- f. The training process of the AI **SHOULD** take less than 3 hours, to allow a faster uptime if the AI needs to retrain after an update.
- g. AI **SHOULD** use card data from the The Hearthstone API on RapidAPI if needed, to collected data such as card effects, images and tribal tags based of the name of the cards.

- h. AI **SHOULD** use deck data scraped from Hearthpwn to train from using a coded web scraper.
- i. Creation of a deck after training **SHOULD** not take more than 10 minutes. As to allow the to be quick to use by others whilst allowing for testing time with the simulator and the generation of graphs
- j. **COULD** generate graphs of performance of decks outputted by the AI and simulators to visualise the training and testing process including the results of the created decks.

## 3.4 Approach

The project was designed using a agile methodology, equipped with logging, initially from a Trello board for the literature review and then migrated to github commit logs for the implementation of the project. The aim was to implement a requirement, testing it, fixing bugs and integration issues and adding additional features, this process was then repeated for each requirement until completion.

Testing was done on a per feature basis, once a feature was implemented, a series of tests examples were created to ensure the functionality of it, if any errors or problems arose the focus would be shifted to fixing that issue. Evaluation was done after testing, once deemed ready, the AI was executed 10 times to create decks for each class which were evaluated over 100 games across all 10 classes, graphs then display these results.

## 3.5 Libraries

This section describes what libraries have been used in this project, the aim is to show transparency and understand some more ambiguous parts of the implementation. Note that all of these libraries are Python libraries.

### 3.5.1 Web Scaper and API

#### 3.5.1.1 Requests

The requests library is a simple HTTP library that allows the user to send HTTP requests, whether it be GET or POST. This library was selected above others because over 500,000 others use this library aswell meaning that the library is stable and enables long term projects to stay active. This is used for both web scraping and api calls as both require to get a uri to get the information needed for the datasets.

### 3.5.1.2 Json

The json library allows for the encoding and decoding of json data, this is used for creating the outputs of the API calls and Web scraping, it is also used as an input for loading these data files and processing the data.

### 3.5.1.3 Beautiful Soup

Beautiful Soup is a library for web scraping, it pulls out data from HTML and XML files. There are three main libraries for web scraping, Selenium, Scrapy and Beautiful Soup. Selenium is mainly used for AJAX requests and Web Application testing, so it is not ideal for this project as the request library is being used to make http calls. Scrapy whilst having more features, and being faster than Beautiful Soup, it is meant for complex web systems with middleware which are not needed in the scope of my project. Beautiful Soup was chosen for its simplicity and flexibility on small scale projects, since this will mainly be used for scraping data to create datasets. Originally Beautiful Soup was used for the collection of card data, it is no longer used for that as the resulting output was missing crucial data such as class, cost and image links, so an API for collecting the card data was used instead, which allowed the data to be stored in JSON format.

## 3.5.2 GAN

### 3.5.2.1 Keras

Keras is a deep learning API that runs using Tensorflow, a machine learning platform which focuses on enabling speedy experimenting, it is simple to use and very well documented making it easy to implement models. This is what was used for the creation of the Generative Adversarial Network, specifically the implementation of the generator and the discriminator.

### 3.5.2.2 Numpy

Numpy is a python library designed for working with arrays, it aims to provide array objects that are 50x faster than standard python lists, since the GAN will be using a lot of arrays this is ideal for achieving faster training times. This library was used throughout the training process of the GAN, from sample generation to final deck results.

## 3.5.3 Evaluation

### 3.5.3.1 Matplotlib

Matplotlib is a data visualization library which allows data to be displayed through graphs and charts, this is used throughout the evaluation process of the project.

### 3.5.3.2 Time

Time is a python library that is used to measure execution time of functions, in this project it is used to calculate execution time of the GAN training and testing process.

### 3.5.3.3 Fireplace

Fireplace is the python library for the Hearthstone simulator, this is used for testing the generated decks against ones from the dataset. There are two other simulators worth considering, Sabberstone and Hearthbreaker. Hearthbreaker was used by the Google Deep Mind team for card generation however the project is no longer maintained, it has been for many years so that one cannot be used to fulfil our requirements. Sabberstone is written has a full interface but for testing that creates unnecessary bloat, so Fireplace is the better choice, it is maintained, and plays games in seconds, the main downside however is that only about half of the cards in the game are implemented. This is because some cards have tricky mechanics that are hard to implement and that the developers are to write interactions manually.



## Bibliography

- [1] J. Isaacs, “Academic writing and the literature review. honours project. robert gordon university, 14th of october,” 2020, [Accessed: 30/10/20]. [Online]. Available: [https://liverguac-my.sharepoint.com/:v/g/personal/j\\_p\\_isaacs\\_rgu\\_ac\\_uk/EWsECIDDZHNDuCcAMkCajlEBg513KeDA7VYZjzDdqZIerg?e=ihykeq](https://liverguac-my.sharepoint.com/:v/g/personal/j_p_isaacs_rgu_ac_uk/EWsECIDDZHNDuCcAMkCajlEBg513KeDA7VYZjzDdqZIerg?e=ihykeq)
- [2] “Hearthstone.” [Online]. Available: <https://playhearthstone.com/en-us>
- [3] “What is skill and luck - multiplayer discussion - hearthstone forums.” [Online]. Available: <https://us.forums.blizzard.com/en/hearthstone/t/what-is-skill-and-luck/4415>
- [4] CCGer, “Deck building vs skillfull play. 29th december.” 2011, [Accessed: 27/10/20]. [Online]. Available: <https://www.mtgsalvation.com/forums/magic-fundamentals/magic-general/327490-deck-building-vs-skillfull-play>
- [5] F. De, M. Silva, M. C. Fontaine, R. Canaan, J. Togelius, S. Lee, and A. K. Hoover, “Evolving the hearthstone meta.” [Online]. Available: <https://github.com/HearthSim/SabberStone>
- [6] R. Robertson, “Jungian archetypes: Jung, gödel, and the history of archetypes,” 2016. [Online]. Available: [https://books.google.fr/books?hl=en&lr=&id=tLJgDAAAQBAJ&oi=fnd&pg=PT7&dq=jungian+archetypes&ots=56eqtqa6K.&sig=Tmk2iRpe1XHy9l\\_kFD3Yc.N-Gnc&redir\\_esc=y#v=onepage&q=jungian%20archetypes&f=false](https://books.google.fr/books?hl=en&lr=&id=tLJgDAAAQBAJ&oi=fnd&pg=PT7&dq=jungian+archetypes&ots=56eqtqa6K.&sig=Tmk2iRpe1XHy9l_kFD3Yc.N-Gnc&redir_esc=y#v=onepage&q=jungian%20archetypes&f=false)
- [7] J. Judlick, “Identifying deck archetypes - articles - tempo storm.” [Online]. Available: <https://tempostorm.com/articles/identifying-deck-archetypes>
- [8] Gamepedia, “Standard format,” 2020, [Accessed: 04/11/20]. [Online]. Available: [https://hearthstone.gamepedia.com/Standard\\_format](https://hearthstone.gamepedia.com/Standard_format)
- [9] G. Zuin and A. Veloso, “Learning a resource scale for collectible card games,” vol. 2019-August. IEEE Computer Society, 8 2019.
- [10] A. Stiegler, C. Messerschmidt, J. Maucher, and K. Dahal, “Hearthstone deck-construction with a utility system.” Institute of Electrical and Electronics Engineers Inc., 5 2017, pp. 21–28.

- [11] B. Entertainment, “Celebrating 100 million players!” 2018, [Accessed: 05/11/20]. [Online]. Available: <https://playhearthstone.com/en-us/news/22636890>
- [12] K. T. Howard, “Free-to-play or pay-to-win? casual, hardcore, and hearthstone,” *Transactions of the Digital Games Research Association*, vol. 4, pp. 147–169, 10 2019. [Online]. Available: <http://todigra.org/index.php/todigra/article/view/103>
- [13] P. Sweetser and J. Wiles, “Current ai in games : a review,” *Australian Journal of Intelligent Information Processing Systems*, vol. 8, no. 1, pp. 24–42, 2002. [Online]. Available: <https://eprints.qut.edu.au/45741/>
- [14] A. K. Hoover, J. Togelius, S. Lee, and F. de Mesentier Silva, “The many ai challenges of hearthstone,” *KI - Kunstliche Intelligenz*, vol. 34, pp. 33–43, 3 2020. [Online]. Available: <https://link.springer.com/article/10.1007/s13218-019-00615-z>
- [15] G. N. Yannakakis and J. Togelius, *Artificial intelligence and games*. Springer International Publishing, 2 2018.
- [16] A. Janusz, T. Tajmajar, and M. Świechowski, “Helping ai to play hearthstone: Aaia’17 data mining challenge,” in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2017, pp. 121–125.
- [17] A. Santos, P. A. Santos, and F. S. Melo, “Monte carlo tree search experiments in hearthstone,” in *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, 2017, pp. 272–279.
- [18] M. Świechowski, T. Tajmajar, and A. Janusz, “Improving hearthstone AI by combining mcts and supervised learning algorithms,” in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, 2018, pp. 1–8.
- [19] T. Back, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, 1996. [Online]. Available: [https://books.google.fr/books?hl=en&lr=&id=htJHI1UrL7IC&oi=fnd&pg=PR9&dq=evolutionary+algorithms&ots=fBl\\_1QSCiT&sig=g5AzYmN078gAvTVrxhGqfPW-Ieg&redir\\_esc=y#v=onepage&q=evolutionary%20algorithms&f=false](https://books.google.fr/books?hl=en&lr=&id=htJHI1UrL7IC&oi=fnd&pg=PR9&dq=evolutionary+algorithms&ots=fBl_1QSCiT&sig=g5AzYmN078gAvTVrxhGqfPW-Ieg&redir_esc=y#v=onepage&q=evolutionary%20algorithms&f=false)
- [20] D. Kowalczyk, “Writing research questions: Purpose and examples,” 2013, [Accessed: 07/11/20]. [Online]. Available: <https://study.com/academy/lesson/writing-research-questions-purpose-examples.html>
- [21] Flipperbw, “Simple hearthstone logging - See your complete play history without TCP, screen capture, or violating the TOS,” 2014, [Accessed: 08/11/20]. [Online]. Available: [https://www.reddit.com/r/hearthstone/comments/268fkk/simple\\_hearthstone\\_logging\\_see\\_your\\_complete\\_play/](https://www.reddit.com/r/hearthstone/comments/268fkk/simple_hearthstone_logging_see_your_complete_play/)

- [22] E. Bursztein, “I am a legend: hacking hearthstone using statistical learning methods,” 2016, pp. 1–8. [Online]. Available: <https://elie.net/static/files/i-am-a-legend-hacking-hearthstone-using-statistical-learning-methods/i-am-a-legend-hacking-hearthstone-using-statistical-learning-methods-paper.pdf>
- [23] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” 1994.
- [24] M. C. Fontaine, F. D. M. Silva, S. Lee, J. Togelius, L. B. Soros, and A. K. Hoover, “Mapping hearthstone deck spaces through map-elites with sliding boundaries.” Association for Computing Machinery, Inc, 7 2019, pp. 161–169. [Online]. Available: <https://dl.acm.org/doi/10.1145/3321707.3321794>
- [25] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*. Springer Berlin Heidelberg, 2015. [Online]. Available: <http://link.springer.com/10.1007/978-3-662-44874-8>
- [26] D. Whitley, “A genetic algorithm tutorial,” *Statistics and Computing*, vol. 4, pp. 65–85, 6 1994. [Online]. Available: <https://link.springer.com/article/10.1007/BF00175354>
- [27] J. J. Merelo, F. Liberatore, A. F. Ares, R. García, Z. Chelly, C. Cotta, N. Rico, A. M. Mora, and P. García-Sánchez, “There is noisy lunch: A study of noise in evolutionary optimization problems,” in *2015 7th International Joint Conference on Computational Intelligence (IJCCI)*, vol. 1, 2015, pp. 261–268.
- [28] S. J. Bjørke and K. A. Fludal, “Sverre johann bjørke knut aron fludal deckbuilding in magic: The gathering using a genetic algorithm,” 2017. [Online]. Available: <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2462429>
- [29] P. Garcia-Sanchez, A. Tonda, G. Squillero, A. Mora, and J. J. Merelo, “Evolutionary deckbuilding in hearthstone,” vol. 0. IEEE Computer Society, 7 2016.
- [30] P. García-Sánchez, A. Tonda, A. M. Mora, G. Squillero, and J. J. Merelo, “Automated playtesting in collectible card games using evolutionary algorithms: A case study in hearthstone,” *Knowledge-Based Systems*, vol. 153, pp. 133 – 146, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950705118301953>
- [31] A. Burkov, *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019. [Online]. Available: <https://books.google.fr/books?id=0jbxwQEACAAJ>
- [32] 3blue1brown. (2017) But what is a neural network? — deep learning, chapter 1. [Accessed: 25/10/20]. [Online]. Available: <https://www.youtube.com/watch?v=aircAruvnKk>

- [33] F. Bre, J. Gimenez, and V. Fachinotti, “Prediction of wind pressure coefficients on building surfaces using artificial neural networks,” *Energy and Buildings*, vol. 158, 11 2017.
- [34] H. N. Ward, D. J. Brooks, D. Troha, B. Mills, and A. S. Khakhalin, “AI solutions for drafting in Magic: the Gathering,” 9 2020. [Online]. Available: <http://arxiv.org/abs/2009.00655>
- [35] J. Jakubik, “A neural network approach to hearthstone win rate prediction,” in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2018, pp. 185–188.
- [36] Q. H. Vu, D. Ruta, A. Ruta, and L. Cen, “Predicting win-rates of hearthstone decks: Models and features that won aiaa’2018 data mining challenge,” in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2018, pp. 197–200.
- [37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: <https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>
- [38] I. J. Goodfellow, “NIPS 2016 tutorial: Generative adversarial networks,” *CoRR*, vol. abs/1701.00160, 2017. [Online]. Available: <http://arxiv.org/abs/1701.00160>
- [39] R. Rodriguez Torrado, A. Khalifa, M. Cerny Green, N. Justesen, S. Risi, and J. Togelius, “Bootstrapping conditional gans for video game level generation,” in *2020 IEEE Conference on Games (CoG)*, 2020, pp. 41–48.
- [40] A. Sehgal, H. La, S. Louis, and H. Nguyen, “Deep reinforcement learning using genetic algorithm for parameter optimization,” in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, 2019, pp. 596–601.
- [41] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, “Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning,” 12 2017. [Online]. Available: <http://arxiv.org/abs/1712.06567>
- [42] A. Nigam, P. Friederich, M. Krenn, and A. Aspuru-Guzik, “Augmenting genetic algorithms with deep neural networks for exploring the chemical space,” *arXiv*, 9 2019. [Online]. Available: <http://arxiv.org/abs/1909.11655>
- [43] C. Z. Janikow, “A knowledge-intensive genetic algorithm for supervised learning,” pp. 33–72, 1993. [Online]. Available: [https://link.springer.com/chapter/10.1007/978-1-4615-2740-4\\_3](https://link.springer.com/chapter/10.1007/978-1-4615-2740-4_3)