The role modern artificial intelligence plays in modern games and their development

22/11/2019

A literature review

Christopher Webb

22-Nov-19

Abstract

Since the start of the gaming industry, adversarial behaviour needed to be logically included (programmed) into games to allow for satisfying gameplay, for example, the enemy ghosts in *Pacman* (1980). Since then, AI in games has developed and continued to be at the forefront of AI development. This review looks at foundational and recent literature on the matter and finds how the current state of machine learning is being utilised in the current games industry in terms of development. It finds that there is a desire to implement this technology but due to the experimental nature, there is no specific way in which these different AI technologies can be applied, finding that most applications were in an attempt to find uses for the new technology. This review also makes recommendations for further diversifying implementations and mass implementations as further research.

Word count: 4, 110 words

Table of Contents

Introduction

Since the beginning of stored computer functions, computer systems have been able to store memorized procedures and execute them as needed. Over time, this was extended to emulate behaviour forming the basis of what would become known as artificial intelligence (AI). AI, as it stands today can be used for a multitude of purposes and can form intelligence in a variety of ways.

In researching for my major project, which was the implementation of a recommendation system utilising machine learning cosine similarity AI to best serve users based on behavioural preferences, I found that the term 'machine learning' was used to describe the general foundations of recent types of AI, and I wanted to explore these other types and applications further. Specifically, I wanted to explore AI in terms of how it works in the many different forms it takes, what types of methods were used to gather intelligence, how more modern implementations were used to improve areas where more primitive behaviour had been utilised and what unique uses of modern AI are. During this research I was reminded of the wider use of AI and its derivatives as they are used in the video games industry – a personal interest area.

Video game design and development is an area where the aforementioned stored computer functions needed to be extended to form behaviours that emulate some aspect of intelligence from the industry's infancy. This has taken many forms in the industry as many roles have some dependency on behaviour. These forms are, but not limited to, the behaviour of NPCs[1], the realism involved in digital worlds, the fidelity of the conveying medium (audio/visual/haptic) as well as roles associated with the design of various aspects of games during their development, for example, reinforcing design choices or generation of assets.

[1] Non-playable character.

Literature Review

Background of Modern Artificial Intelligence

Modern machine learning can be separated into three main learning types: statistical, reinforced and neural networking (plus a fourth not in scope) (See Appendix A). While there are other forms, most techniques fall under these three methods. Key to understanding how these types work, is to understand how they are based on foundations of statistical science.

Supervised learning is a concept derived from statistical analysis in the 1950/60s, wherein two main types of results can be produced: classification results and regressive results, as per the statistical definitions. Most mathematical definitions align to explain classification as "typically represent(ing) a set of problems whereby the goal is to create a predictive model that can discriminate between various known classes… examples of classification datasets (are) where the goals are to identify the correct label for each image" (Dufourq & Basset, 2017). Reinterpreted, classification models can be used to predict data with access to historical data and guidance in training. Supervised classification algorithms include 'lazy learner' – algorithms that "wait to see the test sample before learning a classifier" (Garcia, Feldman, Gupta & Srivastava, 2010), and algorithms like 'relationship distance tests', e.g. K-Nearest Neighbour[2] and Cosine Similarity[3]. In comparison to this, 'eager learner' algorithms, like decision trees[4] and Naïve Bayes[5] "construct a representation of the target function from training instances" (Mazinani & Fathi, 2015) and then "classification of new instances is usually done by rules that use the model" (Mazinani & Fathi, 2015). As such, lazy learning predictions result from the raw algorithm and test data, whereas eager learning predictions result from a model created by the algorithm of the test data. The algorithms mentioned are far from the only ones that exist, but are some of the most used. Dufourq & Basset (2017) go on to state that regression "predictive output is continuous (as opposed to discrete in the case of classification) " with their example being in predicting median house values in Boston, defining the concept as any model predicting a non-binary result.

Conversely, unsupervised learning is the prediction of unlabelled data "without providing an error signal to evaluate the potential solution" (Sathya & Abraham, 2013) with most techniques associated with pattern recognition in a general context or, given that in pattern recognition data being analysed may be n-dimensional in nature, "high-dimensional data often contain measurements on uninformative or redundant variables" (Nguyen & Holmes, 2019). In basic terms, analysis with 'higher dimensions than needed' can produce noise in results. The main application of unsupervised

---

[2] "A simple nonparametric procedure for the assignment of a class label to the input pattern based on the class labels represented by the closest (say, for example, in the Euclidean sense) neighbours of the vector" (Keller, Gray & Givens, 1985)

[3] "Similarity or dissimilarity angle "of n-dimensional vectors where the "cosine of 0° is 1, and it is less than 1 for any other angle." (Zaware, Gautam, Nashte & Khanuja, 2015). Mathematically speaking, cosine similarity is the dot product of the n-dimensional vectors created by datasets.

[4] A tree "represents a segmentation of the data that is created by applying a series of simple rules" where "each rule assigns an observation to a segment based on the value of one input" and "one rule is applied after another, resulting in a hierarchy of segments within segments" (Quadril & Kalyankar, 2010) with a forest being made up of many trees.

[5] Naïve Bayes algorithms "assumes that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the class variable" (Mazinani & Fathi, 2015).

learning is clustering, that is, categorization of “similar data into one cluster based on some similarity measures” (Min et al., 2018), and is used more as a research method where the labels are unknown and a well-known algorithm K-Mean[6] is used. However, due to the disadvantages involved, specifically “its lack of robustness to noise, high dimensionality, and outliers” (Nikulin & McLachlan, 2009), mean shift clustering[7] has been a popular alternative.

Reinforcement learning, wherein “examples (data) are not carefully selected by a teacher… instead, the distribution of examples is influenced by the robot's actions, since the states and rewards experienced by the robot depends on the actions it takes” (Mahadevan, 1996) when applied to robotics, is a method closely emulating supervised learning except that labelled data may not partially or fully exist. While there are many methods of reinforcement learning, the most common is typically Q-Learning[8]. While reinforcement learning has not been used often in a more historical context, understanding it and supervised/unsupervised learning can act as an entry point into modern machine learning: artificial neural networks.

Artificial neural networks[9] are machine learning algorithms that have layers of nodes or ‘neurons’ that perform smaller calculations of the input data, passing the results to the next ‘node’ in line. In most cases today, there are ‘hidden layers’ between the input and output layers. Such AI designs are modelled to emulate the human brain with deep learning (DL) being “about accurately assigning credit across many such stages” (Schmidhuber, 2015), meaning that there are more layers than usual, hence the term ‘deep’. As deep learning AI cover a multitude of areas out of scope, the main focus in this review will be CNNs and GANs as they are two of the most popular deep learning types.

CNNs (convolutional neural networks) are a type of deep learning that is “designed to process data that come in the form of multiple arrays” (LeCun, Bengio & Hinton, 2015) according to most industry leaders with common examples being “1D for signals and sequences, including language; 2D for images or audio spectrograms; and 3D for video or volumetric images” (LeCun et al., 2015). The process works, typically, by analysing dimensional data, for example, an image (a 2D array of 4D data: red, green, blue and alpha values), and looking for patterns via nodes containing filters, for example, with matrix multiplication using edge detection filters (MicroImages, 2014), with further basic patterns being lines, circles and other geometric features. An example of a CNN in use is Google’s ‘deep dream’ project (Rayner, 2016). By analysing n-amounts of patterns via the neural network chain, classification can be made on the otherwise complicated input data.

[6] K-mean “can be thought of as a gradient descent procedure, which begins at starting cluster centroids”, then, “iteratively updates these centroids to decrease the objective function” and will “always converge to a local minimum” (Oyelade, Oladipupo & Obagbuwa, 2010).

[7] A mean shift clustering “is to treat the points in the d-dimensional feature space as an empirical probability density function where dense regions in the feature space correspond to the local maxima or modes of the underlying distribution.” (Derpanis, 2005).

[8] “Q-learning works by incrementally updating the expected values of actions in states. For every possible state, every possible action is assigned a value which is a function of both the immediate reward for taking that action and the expected reward in the future based on the new state that is the result of taking that action” (Gaskett, Wettergreen & Zelinsky, 1999).

[9] A “neural network consists of many simple, connected processors called neurons, each producing a sequence of real-valued activations. Input neurons get activated through sensors perceiving the environment, other neurons get activated through weighted connections from previously active neurons” (Schmidhuber, 2015), the most common type of process of which is a ‘feed forward’ process where nodes cascade outputs forward through the network, whereas recurrent processes have at least a single loopback involved.

Lastly, GANs (generative adversarial networks) are recent developments in AI technology where combinations of types of AI are partnered in an adversarial manner to achieve a generative goal. They can be thought of as a generative AI "pitted against an adversary: a discriminative model that learns to determine whether a sample is from the model distribution or the data distribution" (Goodfellow et al., 2014) where the result from the discriminator[10] is fed back to the generator as input which moulds the generator's output to try to pass the discriminator, a common analogy is "counterfeiters trying to produce fake currency and use it without detection" and "police, trying to detect the counterfeit currency" (Goodfellow et al., 2014). It is important to note that GANs are an example of semi-supervised and combined machine learning as well as that, with GANs, two sets of classification are provided, information on how to identify patterns and how to create them.

[10] Typically a CNN for pattern recognition (Barua, Erfani & Bailey, 2019). For example, if a CNN looking for patterns indicative of a human face, the discriminator will classify based on whether or not the generator output something looking similar to a face.

Primitive Artificial Intelligence in Game Design/Development

In areas where possible (not necessarily in all mentioned areas due to computational limits at the time), developers attempted to implement AI to improve the quality of games. Video games are dependent on some form of immersion, so the ability of NPCs to behave believably and clearly in such a way that is interpreted as, at least, 'near human' is vital. NPCs and bots[11] had rudimentary pseudo-intelligence implemented from early in the industry's history with the first major milestone being the chess playing 'Deep-Blue', which won against Garry Kasparov[12] with the AI implemented being a "minimax algorithm of the ideas of heuristic static evaluation and alphabeta pruning, all of which were developed in the 1950s" (Korf, 1997), where 'minimax' algorithms are primitive statistical functions. While Deep-Blue was not a 'video' game, it was the first major instance of a game having an automated adversary. More examples later included arcade games like Pacman, where each ghost has a unique 'personality', or "behaviour that supports multiple interpretations, giving the ghosts an 'inner life'" (Mateas, 2003), as a 2003 paper explains, noting that the game depends on the enemy behaviour to make the game, as "without the ghosts there is no challenge…and hence no game" (Mateas, 2003).

As the games industry continued to evolve, the next major step in AI was finite state machines (FSMs) in the 1990s which are ultimately, in tandem with machine learning, used today. A FSM, as described by the NIST, "models a machine controller as being in one of a finite set of the possible states, known as the state-space, at any given time" (Michaloski, Proctor & Rippey, 2003). In basic interpretation, FSMs are a set of interconnected states representing behaviours where there are changes based on triggers and variables, for example, a guard in a stealth game looking for the player after losing sight of them, only to fail to find them after a set amount of time and return to a relaxed state. "FSMs are used more frequently in computer games than any other AI technique" (Sweetser & Wiles, 2002) and a modern example of FSMs in games is that of *Alien: Isolation* (2014), in which the FSM AI benefits from "a decision making process that takes into consideration the player's behaviour and also learns from it (and) can create a more immersive world" (de Lima & Madeira, 2017).

Alien: Isolation has a both 'active' and 'passive' 'micro' and 'macro' FSM (Thompson, 2016), that creates an experience in which "players (need) to complete a task or overcome an AI opponent where the AI is aiming to create an experience… rather than defeat the player" (Treanor et al., 2015), or 'AI as Villain' in other terms. Other examples of FSM are cases where games are used in different design patterns like 'AI as Adversary' (e.g. chess or *Rocket League* (2015)) and 'AI as Role-model' (e.g. *Spy Party* (2009), where players must, as best as they can, emulate the behaviour of FSM bots) (Treanor et al., 2015). With FSMs leading AI in videogames, a decades old concept has started to be pushed to limits as FSMs "tend to be poorly structured with poor scaling… they increase in size uncontrollably as the development cycle progresses… FSM maintenance (is) very difficult... FSMs in games tend to include states within states, multiple state variables, randomness in state transitions and code executing every game tick within a state" (Sweetser & Wiles, 2002). The

[11] Bots are programmatic 'stand ins' to a character that would otherwise be controlled by a player, for example, in a racing game without a second player to act as a second driver of a second car, if the game was designed to have at least two cars racing, the game would implement a racing bot in the second car.
[12] A world chess champion at the time.

disadvantages of FSMs, as games grew in computational complexity (UMass Amherst, 2014) over the mid-2000s, led to the desire for more modern AI in games and possibly the development process.

Modern Artificial Intelligence in Game Design/Development

With modern developments in machine learning (ML) technology, the application of ML AI can be utilised in many ways throughout the development process, affecting AI characters, realism, fidelity and design.

Where FSMs have been used mostly in the past in terms of bot and NPC behaviour, more games are now using machine learning to provide players with relatively real opponents. One such experimental example is *Starcraft* (1998), a RTS[13] game in which ML was implemented in 2011 for "neural networks to learn the expected reward for attacking or fleeing with a particular unit in a given state, and chose the action with the highest expected reward when in-game" (Robertson & Watson, 2014); essentially, a supervised, reinforced neural network AI to combat players. This was later, in essence, implemented in *Starcraft 2* (2010) in 2019 as AlphaStar, as an AI "initially trained using imitation learning to mimic human play, and then improved through a combination of reinforcement learning and self-play". Similar systems have also been implemented for *Dota 2* (2013) (Arulkumaran, Cully & Togelius, 2019), *Planetary Annihilation* (2014) (de Matos, 2014), *Total War: Warhammer* (2016) (Barriga, Stanescu, Bosoain & Buro, 2019), experiments involving automatic playing of *Super Mario World* (aleju, 2016) as well as in racing games as testing methods for self-driving car research (DeepRacing, n.d.) and universally applied, like Deepmind's AlphaGo (Hölldobler, Möhle & Tigunova, 2017).

Another area recently being associated with ML is realism in video games. An example of this is in animations, which due to limitless possible movements in real life must, for the sake of development, be downscaled to a FSM. Recently, in paper written about neural networks in animation, a method wherein key-frames[14] of motion capture data are interpolated between states by a system[15] that "can handle a wide variety of motion types and allows the users to guide the characters by only providing abstract goals" (Starke, Zhang, Komura & Saito, 2019) by predicting and interpolating between key frames based on regressive probability is explained. Another aspect of realism in development is maintenance after launch, for example, in a multiplayer game, enforcing integrity. Bots have a history of being used by players to cheat[16] and to combat this, Valve, for example, have implemented deep learning detection for cheating players of their game *Counter-strike: Global Offensive* (2012) by matching patterns consistent with probable 'aim-bot' behaviour[17] (McDonald, 2018).

Efforts to attain some level of realism have always been sought after in video games (with methods outside of AI like anti-aliasing methods and lighting models), so have efforts to maximise and

---

[13] Real time strategy usually associated with the real time allocation of units from a 'war-room perspective' attempting to 'out plan' enemies.

[14] The animation term used to describe a discrete key moment of specific movement.

[15] The "Motion Prediction Network is composed of two modules where the first module is the encoder module that receives the components of the character state in the previous frame and encodes them individually using simple three-layer networks. The second module is the prediction module that receives the output of the encoder module and predicts the state of the character in the current frame." (Starke et al., 2019)

[16] Mainly used in FPS games, used to react faster to competition, client-side primitive AI known as 'aim-bots'. (Mascellino, 2019).

[17] An issue in researching this is that companies cannot be transparent in how these systems work to protect the integrity of the implementation, relaying how the implementation works would compromise this, something actually mentioned in the McDonald GDC talk (McDonald, 2018).

synthesise fidelity. In recent advancements, using deep learning, Nvidia[18] has partnered with Microsoft to allow DirectX 12[19] support of deep learning super sampling via the NGX architecture (Nvidia, n.d.). Andrew Edelsten[20] explains that Nvidia gather video game "frames from the target game, and then for each one we generate a matching 'perfect frame' using either super-sampling or accumulation rendering" (Edelsten, 2019) where a "supercomputer trains the DLSS model to recognize aliased inputs and generate high quality anti-aliased[21] images that match the 'perfect frame' as closely as possible" (Edelsten, 2019). This utilizes technology described in depth in a 2019 paper on super resolution where CNN and GAN "algorithms have demonstrated great superiority to reconstruction-based and other learning-based methods" (Yang et al., 2019) where it is possible to scale lower resolution images up and predict the missing information using neighbouring pixels, creating a higher resolution image from a low resolution. Nvidia's approach is to utilise this to "boost frame rates at high GPU workloads" (Edelsten, 2019) which is a use in line with the conclusion of a 2017 paper on the matter where the "approach can achieve state-of-the-art results on standard benchmarks with a relatively high speed" (Jia, Xu, Cai & Guo, 2017). There are a number of derivative algorithms dependant on different neural network setups, with some better than others due to 'GPU implementation' over 'CPU implementation' (Kim, Lee & Lee, 2016), with Nvida's being a GPU implementation. While the above improvements brought about by ML and DL have been indirectly assisting game development by alleviating the need for developers to, for example, manually catch cheaters, program all standard logic for bots and NPCs to behave as if they were human, instead, allowing them to learn how to act through various means, and allowing for versatile rendering, in the case of Nvidia's DLSS, to allow for more development in visual performance over visual optimisation, these do not play an active part in the development cycles; however, with recent advancements in GANs, this is could start to change.

With games that require procedural generation[22], like *Minecraft* (2009) with the procedurally generated world on the start of a new game, values of assets are generated based on hashing algorithms. There are issues with this however as there is a limit, depending on what algorithm is used, to how much variability can be introduced as well as, while it can be weighted, mainly being random or all assets used in a game needing to be produced manually. This is an area that GANs are recently showing promise in and could possibly be used in the development process. Recently, *Doom* (1993), due to its open source nature, had a fan-made tool utilising GANs to generate levels. The tool "does not require an expert to encode it explicitly, as traditional procedural generation often does" (Giacomello, Lanzi & Loiacono, 2018) where a "generator network receives in input… (of) six images extracted from the level WAD[23] file a vector of level features, and a noise vector sampled from a Gaussian distribution" (Giacomello et al., 2018) with the goal of the tool to "highlight a viable alternative to classical procedural generation" (Giacomello et al., 2018). A second example of this is using GANs in a similar way to make dungeons in Zelda. A recent paper on this suggests "bootstrapping... (AI) trained on just five human-designed levels" (Torrado et al., 2019) to possibly

[18] A leading GPU manufacture.
[19] The most recent graphics rendering API version owned by Microsoft.
[20] Technical Director of Deep Learning at NVIDIA (Edelsten, 2019)
[21] Aliasing is a visual artifact on diagonally rendered lines where the 'stair stepping' of the pixel grid creates a jagged line instead of a smooth one, anti-aliasing is a term for methods that attempt to resolve this.
[22] Automatic generation of an asset of some sort.
[23] WAD file, 'Where's All the Data' file, *Doom* developer *ID Software*'s proprietary file format.

aid development efforts – something the industry leaders are now heading towards prototypically (del Val, 2019).

Discussion

As mentioned above, some industry leaders have started to look into implementing GANs including Electronic Arts (del Val, 2019), a company that owns a large portion of the intellectual property in the games industry and therefore has a large amount of total development time per annum. In an interview, the technical director of SEED[24] stated that the short-term goal for deep learning AI was to "help the studio to collect more crash reports and find more bugs" but eventually to be "engaging with human players" (Electronic Arts, n.d.). It seems there is such a compelling reason to utilise AI wherever possible, yet the industry has not adopted it in full force.

A reason for this is that ML does have disadvantages. On a case-by-case basis, ML algorithms are rarely applicable, "Every algorithm has some of drawbacks" (Novak, Čep & Verber, 2017), or in some cases needing derivatives of the same algorithm, which is the case with a GAN research article where it is stated that, in that application, GANs by themselves "have difficulty in generating playable and unique levels when few training samples are available" (Torrado et al., 2019). Another reason for this lack of adoption is the idea of it being a fairly new technology computationally. In the context of gaming, the only consumer hardware in the space capable of deep learning is the 'RTX' range of GPUs by Nvidia (the only of Nvidia's range to support the DLSS feature mentioned) released in 2018 (Chacos, 2018) at a premium price (Freedman, 2018), exclusive to PC players, wherein the tensor core[25] was not even the main feature. It seems that in the current gaming space, on the consumer side, there is not a high demand for DL. This is particularly an issue given that most game development studios use consumer hardware (Lohman, 2010). ML, while a new technology that seems experimental to large companies, is financially out of reach for many smaller developers as they have no full access to the technology and, below a certain size, studios have no access to this technology from the vendor outside of consumer technology.

While there are financial and conceptual disadvantages to smaller developers, commonly used game engines[26] are supporting their users with tools for ease of access via accessing APIs (like Tensorflow[27]'s) that allow for DL work. For example, Unity Technologies, in 2017, announced the 'ML agents' platform for Unity stating "using the Unity Editor… intelligent agents can be trained using Deep Reinforcement Learning, Evolutionary Strategies, or other machine learning methods through a simple to use Python API" (Juliani, 2017).Similar work has been done in Epic Games' Unreal Engine using the 'blueprint' scripting system (Boyd, 2017).

The four development aspects mentioned in this review could play a big role in game development in the foreseeable future with the potential for rapid expansion in complexity and realism in the development landscape. For example, AI has started to play roles in development and "creates emergent gameplay[28]" (Cicala, 2017). "(AI) never reached its full potential because the developers

---

[24] SEED, Electronic Arts' experimental AI division

[25] The processor associated with the best deep learning as it is able to apply the specific calculations optimally.

[26] Game engines are the development suites that are associated with the making of the assets and the scripting of the code that goes into a game.

[27] An open source ML platform created by Google.

[28] Emergent gameplay is gameplay that is dynamic, happening when complicities in a game world work together to create a unique experience.

needed to hold parts of it back, but its overall effect on the game was clearly noticed" (Cicala, 2017) as mentioned in an article of the AI development of game series know as *STALKER,* explaining that the bots and NPCs in the STALKER world need uniqueness and realism, a sentiment relevant to the concept of "enhanced realism also makes the player feel more involved in the game" (Wake Forest University, 2010). This realism is something felt in the ESports community as AI in RTS and MOBA[29] games are now being used for training competitors at a professional level (Ozer, 2019), Starcraft is an example of this.

Graphical fidelity is now, due to computational ability, capable of reaching 'holy grails' once thought impossible (Cardinal, 2018) realistically, the same "photorealistic high-end visualization paradigm (that) will be an ideal for some time" (Masuch & Röber, 2005) . With 'high-end visualisation' more of an important factor in games, this is directly affects the required effort needed by developers. One area of possible gain is in the use of GANs, for example, through MITs recent experiment with 'video to video' synthesis where video can be completely transformed into different contexts (for example, a video of a landscape in good weather can be reimagined as being in a blizzard) (Wang et al., 2018). Applied to game development, the ability for DL to reduce developmental needs on the realism and visual fidelity (as MITs experiments handle high resolution data) could potentially dramatically reduce development needs and costs to create the same high standard (Thomas, 2019).
While these are direct methods that may be implemented in the development process, there are features like DLSS that may be helpful post-launch both currently and potentially (for example, potential use with 'checkerboard rendering' as a reconstruction technique (Andersson, 2018)).

However, aside from NPC behaviour, the largest potential role ML could play in game development is in assisting the development and design process. In the *Doom* and *Zelda* level generation papers, the AI had an active role designing maps[30] for the researchers. Translated to actual development, had this technology existed in the industry at that time, it could have dramatically changed how development took place, particularly given the Doom level designer's outlook on the idea of programming and art (Ewalt, 2006) suggesting the idea would be embraced. However, there would then be a question of 'if CNNs and GANs can be capable of creative work, what does creativity become?' This is one of many questions pertaining to a new direction for the industry. As well as this, as a lot of AI calculations are done on GPU's (for their ability to parallel compute), another question at hand is 'how can other areas of AI assist in development?' For example, recent advancements in 3D fluid simulations (Kim et al., 2019) (which until recently could not "fulfil the demands for frame rates in interactive environments… (therefore using) simplified physical models in combination with 2D visual approximations like sprites, billboards or animated images" (Krüger & Westerman, 2005)).

Despite the ambiguity of AI involvement in games development that exists in the current space, it is clear that there is some desire for it and curiosity within the industry for experimentation. It is possible that, once the main and most efficient uses of DL and ML as a whole are found via aforementioned experimentation, the industry can start to widely adopt AI technology in a more developmental role as, currently, it plays a heavily prototypical one.

[29] MOBA, massively online battle area game.
[30] Maps are areas or levels of a game

Conclusion

Future Recommendations and Implications of Research

In other areas such as video and audio development, AI is already starting to become noteworthy. An example of this is ‘deep-fakes’, a concept known for taking the likeness of a person or subject and using GANs to create footage that would not otherwise exist, sometimes being used nefariously (Green, 2019). Future possible research could involve exploring the use of this application for video game development, possibly in areas of dynamic game generation and the generation of a game based on the preferences of the player, much like the idea of recommendation systems.

With algorithms being completely useful only in certain circumstances, it appears that no single algorithm or ML method is best which is why a future possible avenue of research is to explore how best to use each type of AI and better integrate the findings and “strengths of algorithms together… (to) bring us to better classification accuracy” (Novak et al., 2017).

Summary

The findings of this review show that there is a multitude of approaches to ML in the game development space as well as the wider technology industry. As this is a fairly new area of development, much of the use of this technology remains in a malleable flux. With four key areas of video game development (that the recent DL advancements in ML could be/are better utilised in) outlined (those being NPC behaviour, realism, fidelity and assistance in development and design), it has been found that, in all four, there are some applications that can dramatically affect the industry moving forward if it has not been already. Specifically, once the review outlined literature on the fundamentals of modern ML, it explained certain cases where reinforced, supervised and unsupervised learning were used in the games industry relating to developments, then explored the possibility of using artificial neural networks to generate content, in cases of realism, identity and in use during development. The review found and discussed that, while these technologies do have disadvantages, different algorithms and methods of AI should be used for different use cases and as a result, recommended the exploration of more combined use of AI.

References

Aleju. (2016, May 27). Playing Mario with Deep Reinforcement Learning. Retrieved 17 November 2019 from https://github.com/aleju/mario-ai

Andersson, L. (2018). Comparison of Anti-Aliasing in Motion. Retrieved 19 November 2019 from http://www.diva-portal.org/smash/get/diva2:1260863/FULLTEXT02.pdf

Arulkumaran, K., Cully, A. & Togelius, J. (2019). AlphaStar: An Evolutionary Computation Perspective. *GECCO '19 Proceedings of the Genetic and Evolutionary Computation Conference Companion,* 314-315. doi:10.1145/3319619.3321894

Barriga, N. A., Stanescu, M., Besoain, F. & Buro, M. (2019). Improving RTS Game AI by Supervised Policy Learning, Tactical Search, and Deep Reinforcement Learning. *IEEE Computational Intelligence Magazine, 14*(3), 8-18. doi:10.1109/MCI.2019.2919363

Barua, S., Erfani, S. M. & Bailey, J. (2019). FCC-GAN: A Fully Connected and Convolutional Net Architecture for GANS. *ArXiv. Abs/1905.02417.* Retrieved 17 November 2019 from https://arxiv.org/pdf/1905.02417.pdf

Boyd, R. (2017). Implementing Reinforcement Learning in Unreal Engine 4 with Blueprint. Retrieved 20 November 2019 from https://jewlscholar.mtsu.edu/handle/mtsu/5247

Cardinal, D. (2018, August 21). How Nvidia's RTX Real-Time Ray Tracing Works. Retrieved 20 November 2019 from https://www.extremetech.com/extreme/266600-nvidias-rtx-promises-real-time-ray-tracing

Chacos, B. (2018, September 14). Ray traced games won't launch with Nvidia's GeForce RTX graphics cards. Retrieved 20 September 2019 from https://www.pcworld.idg.com.au/article/646762/ray-traced-games-won-t-launch-nvidia-geforce-rtx-graphics-cards/

Cicala, A. (2017, August 15). A-Life: An Insight into Ambitious AI. Retrieved 20 November 2019 from https://gamedevlibrary.com/a-life-an-insight-into-ambitious-ai-2eb435fa4054

de Lima, T. A. & Madeira, C. A. G. (2017). An Investigation of Using Monte-Carlo Tree Search for Creating the Intelligent Elements of Rise of Mitra. *SBC – Proceedings of SBGames.* Retrieved 18 November 2019 from https://www.sbgames.org/sbgames2017/papers/ComputacaoShort/175198.pdf

de Matos, X. (2014, June 6). Meet the computer that's learning to kill and the man who programmed the chaos. Retrieved 19 November 2019 from https://www.engadget.com/2014/06/06/meet-the-computer-thats-learning-to-kill-and-the-man-who-progra/

del Val, J. (2019, March 18-22). Towards Deep Generative Models in Game Development. *Game Developers Conference* [Lecture Slides]. Retrieved 20 November 2019 from https://media.contentapi.ea.com/content/dam/ea/seed/presentations/gdc19-towardsdeepgenerativemodelsingamedevelopment.pdf

DeepRacing. (n.d.). Deep Racing. Retrieved 20 November 2019 from https://deepracing.ai/

Derpanis, K. G. (2005, August 15). Mean Shift Clustering. Retrieved 19 November 2019 from https://pdfs.semanticscholar.org/da1b/8039bd3ea75404369da61846e70326ea9708.pdf

Dufourq, E. & Bassett, B. A. (2017). Automated problem identification: regression vs classification via evolutionary deep networks. *SAICSIT '17 Proceedings of the South African Institute of Computer Scientists and Information Technologists,* (1-9). doi:10.1145/3129416.3129429

Electronic Arts. (n.d.). Teaching AI-Agents to Play Battlefield. Retrieved 20 November 2019 from https://www.ea.com/en-gb/news/teaching-ai-agents-battlefield-1

Ewalt, D. M. (2006, December 14). John Romero On Videogames As Art. Retrieved 19 November 2019 from https://www.forbes.com/2006/12/10/john-romero-games-tech_cx_de_games06_1212romero.html#1911d39a6866

Freedman, A. E. (2018, August 21). Why Are Nvidia's GeForce RTX GPUs So Expensive?. Retrieved 20 November 2019 from https://www.tomshardware.com/news/nvidia-rtx-gpus-price-why-expensive,37672.html

Garcia, E. K., Feldman, S., Gupta, M. R. & Srivastava, S. (2010). Completely Lazy Learning. *IEE Transactions on Knowledge and Data Engineering, 22*, 1274-1285. doi:10.1109/tkde.2009.159

Gaskett, C., Wettergreen, D. & Zelinsky, A. (1999). Q-Learning in Continuous State and Action Spaces. In N. Foo (Eds.), *Advance Topics in Artificial Intelligence* (pp. 417-428). doi:https://doi.org/10.1007/3-540-46695-9_35

Giacomello, E., Lanzi, P. L. & Loiacono, D. (2018). DOOM Level Generation using Generative Adversarial Networks. Retrieved 17 November 2019 from https://arxiv.org/pdf/1804.09154.pdf

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014). Generative Adversarial Nets. *NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems, 2*, 2672-2680. Retrieved 17 November 2019 from https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

Green, A. (2019, October 30). Lawmakers and tech groups fight back against deepfakes. *Financial Times.* Retrieved 21 November 2019 from https://www.ft.com/content/b7c78624-ca57-11e9-af46-b09e8bfe60c0

Hölldobler, S., Möhle, S. & Tigunova, A. (2017). Lessons Learned from AlphaGo. *Proceedings of the Second Young Scientist's International Workshop on Trends in Information Processing,* 92-101. Retrieved 20 November 2019 from https://www.researchgate.net/publication/318825598_Lessons_Learned_from_AlphaGo

Jia, X., Xu, X., Cai, B. & Guo, K. (2017). Single Image Super-Resolution Using Multi-Scale Convolutional Neural Network. In B. Zeng, Q. Huang, A. El Saddik, H. Li, S. Jiang, X. Fan (Eds), *Advances in*

Multimedia Information Processing – PCM 2017. doi:https://doi.org/10.1007/978-3-319-77380-3_15

Juliani, A. (2017, September 19). Introducing: Unity Machine Learning Agents Toolkit. Retrieved 18 November 2019 from https://blogs.unity3d.com/2017/09/19/introducing-unity-machine-learning-agents/

Keller, J. M., Gray, M. R. & Givens, J. A. (1985). A Fuzzy K-Nearest Neighbor Algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-15*(4), 580-585. doi:10.1109/TSMC.1985.6313426

Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M. & Solenthaler, B. (2019). Deep Fluids: A Generative Network for Parameterized Fluid Simulations. *Eurographics, 30*(2). Retrieved 21 November 2019 from https://arxiv.org/pdf/1806.02071.pdf

Kim, J., Lee, J. K. & Lee, K. M. (2016). Accurate Image Super-Resolution Using Very Deep Convolutional Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition.* doi:10.1109/CVPR.2016.182

Korf, R. E. (1997). Does Deep-Blue use AI?. *ICGA Journal, 20*(4), 243-245. doi:10.3233/ICG-1997-20404

Krüger, J. & Westermann, R. (2005). GPU Simulation and Rendering of Volumetric Effects for Computer Games and Virtual Environments. *Eurographics, 42*(3). doi:https://doi.org/10.1111/j.1467-8659.2005.00893.x

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *Nature, 521*, 436-444. doi:doi:10.1038/nature14539

Mahadevan, S. (1996). Machine Learning for Robots: A Comparison of Different Paradigms. *In Workshop on Towards Real Autonomy, IEEE/RSJ International Conference on Intelligent Robots and Systems.* Retrieved 19 November 2019 from https://pdfs.semanticscholar.org/344e/f760c9a16d80a4cfb103721a1228c7c6d9b0.pdf?_ga=2.52213161.1230355086.1574309937-1371438404.1574156489

Mascellino, A. (2019, October 24). Gaming Aimbots: Useful Technology or Ruthless Cheat?. Retrieved 19 November 2019 from https://discover.bot/bot-talk/gaming-aimbots/

Masuch, M. & Röber, N. (2005). Game Graphics Beyond Realism: Then, Now, and Tomorrow. Retrieved 18 November 2019 from http://www.digra.org/wp-content/uploads/digital-library/05150.48223.pdf

Mazinani, S. M. & Fathi, K. (2015). Combining KNN and Decision Tree Algorithms to Improve Intrusion Detection System Performance. *International Journal of Machine Learning and Computing, 5*(6), 476-479. doi:10.18178/ijmlc.2015.5.6.556

McDonald, J. (2018, March 31). GDC 2018: John McDonald (Valve) – Using Deep Learning to Combat Cheating in CSGO [Video File]. Retrieved 19 November 2019 from https://www.youtube.com/watch?v=ObhK8lUfIlc

Michaloski, J. L., Proctor, F. M. & Rippey, W. G. (2003). Expanding the Role of Finite State Machine Technology in Open Architecture Control. *Proceedings of the 2nd CIRP International Conference of Reconfigurable Manufacturing.* Retrieved 18 November 2019 from https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=822516

MicroImages, Inc. (2014). Raster & Image Processing: Edge Detection Filters. Retrieved 17 November 2019 from https://www.microimages.com/documentation/TechGuides/81FiltEdge.pdf

Min, E., Guo, X., Liu, Q., Zhang, G., Cui, J. & Long, J. (2018). A Survey of Clustering With Deep Learning: From the Perspective of Network Architecture. *IEEE Access, 6I, 39501-39514.* doi:10.1109/ACCESS.2018.2855437

Nguyen, L. & Holmes, S. (2019). Ten quick tips for effective dimensionality reduction. *PLoS Computational Biology, 15*(6). doi:https://doi.org/10.1371/journal.pcbi.1006907

Nikulin, V. & McLachlan, G. J. (2009). Regularised k-Means Clustering for Dimension Reduction Applied to Supervised Classification. *Computational Intelligence Methods for Bioinformatics and Biostatistics, 6160*, 82-96, doi:https://doi.org/10.1007/978-3-642-14571-1_7

Novak, D., Čep, A. & Verber, D. (2018). Classification of modern real-time strategy game worlds. *GSTF Journal on Computing (JoC), 6*(1), 1-6. doi:10.5176/2251-3043_6.1.100

Nvidia. (n.d.). NVIDIA NGX Technology – AI for Visual Applications. Retrieved 20 November 2019 from https://developer.nvidia.com/rtx/ngx

Oyelade, O. J., Oladipupo, O. O., Obagbuwa, I. C. (2010). Application of k-Means Clustering algorithm for prediction of Students' Academic Performance. *International Journal of Computer Science and Information Security, 7*(1), 292-295. Retrieved 17 November 2019 from https://arxiv.org/ftp/arxiv/papers/1002/1002.2425.pdf

Ozer, B. (2019, May 9). AI is becoming esports' secret weapon. Retrieved 21 November 2019 from https://venturebeat.com/2019/05/09/ai-is-becoming-esports-secret-weapon/

Quadril, M. N. & Kalyankar, N. V. (2010). Drop Out Feature of Student Data for Academic Performance Using Decision Tree Techniques. *Global Journal of Computer Science and Technology, 10*(2), 2-5. Retrieved 10 November 2019 from https://computerresearch.org/index.php/computer/article/view/891/890

Rayner, A. (2016, March 29). Can Google's Deep Dream become an art machine?. Retrieved 18 November 2019 from https://www.theguardian.com/artanddesign/2016/mar/28/google-deep-dream-art

Robertson, G. & Watson, I. (2014). A Review of Real-Time Strategy Game AI. *AI Magazine, 35*(4), 75-104. doi:https://doi.org/10.1609/aimag.v35i4.2478

Sathya, R & Abraham, A. (2013). Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *International Journal of Advanced Research in Artificial Intelligence, 2*(2), 34-38. doi:10.14569/IJARAI.2013.020206

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks, 61*, 85-117. doi:https://doi.org/10.1016/j.neunet.2014.09.003

Starke, S., Zhang, H., Komura, T. & Saito, J. (2019). Neural State Machine for Character-Scene Interactions. *ACM Trans. Graph, 38*(6). doi:https://doi.org/10.1145/3355089.3356505

Sweetser, P. & Wiles, J. (2002). Current AI in games: a review. *Australian Journal of Intelligent Information Processing Systems, 8*(1), 24-42. Retrieved 18 November 2019 from https://eprints.qut.edu.au/45741/1/AJIIPS_paper.pdf

Thomson, T. (2017, October 31). The Perfect Organism: The AI of Alien: Isolation. Retrieved 16 November 2019 from https://www.gamasutra.com/blogs/TommyThompson/20171031/308027/The_Perfect_Organism_The_AI_of_Alien_Isolation.php

Thompson, T. (2016, August 15). The AI of Alien: Isolation | AI and Games [Video File]. Retrieved 16 November 2019 from https://www.youtube.com/watch?v=Nt1XmiDwxhY&feature=emb_title

Torrado, R. R., Khalifa, A., Green, M. C., Justesen, N., Risi, S. & Togelius, J. (2019). Bootstrapping Conditional GANs for Video Game Development. Retrieved 20 November 2019 from https://arxiv.org/pdf/1910.01603.pdf

UMassAmherst. (2014, February 10). Techbytes: A Brief and Abbreviated History of Gaming Storage. Retrieved 18 November 2019 from https://blogs.umass.edu/Techbytes/2014/02/10/history-of-gaming-storage/

Wake Forest University. (2010, December 22). Video games and realism. Retrieved 20 November 2019 from https://phys.org/news/2010-12-video-games-realism.html

Wang, T., Liu, M., Zhu, J., Lui, G., Tao, A., Kautz, J. & Catanzaro, B. (2018). Video-to-Video Synthesis. *NIPS' 18 Proceedings of the 32nd International Conference on Neural Information Processing Systems,* 1152-1164. Retrieved 20 November 2019 from https://arxiv.org/pdf/1808.06601.pdf

Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J., Liao, Q. (2019). Deep Learning for Single Image Super-Resolution: A Brief Review. *IEEE Transactions on Multimedia*. doi:10.1109/TMM.2019.2919431

Zaware, S. N., Gautam, A., Nashte, S. & Khanuja, P. (2015). An Effectual Approach for Calculating Cosine Similarity. *International Journal of Advance Engineering and Research Development, 2*(4), 13-18. Retrieved 05 November 2019 from http://www.ijaerd.com/papers/finished_papers/AN%20EFFECTUAL%20APPROACH%20FOR%20CALCULATING%20COSINE%20SIMILARITY-25196.pdf

Zubarev, V. (n.d.). Machine Learning for Everyone. Retrieved 17 November 2019 from https://vas3k.com/blog/machine_learning/

Appendix

Appendix A

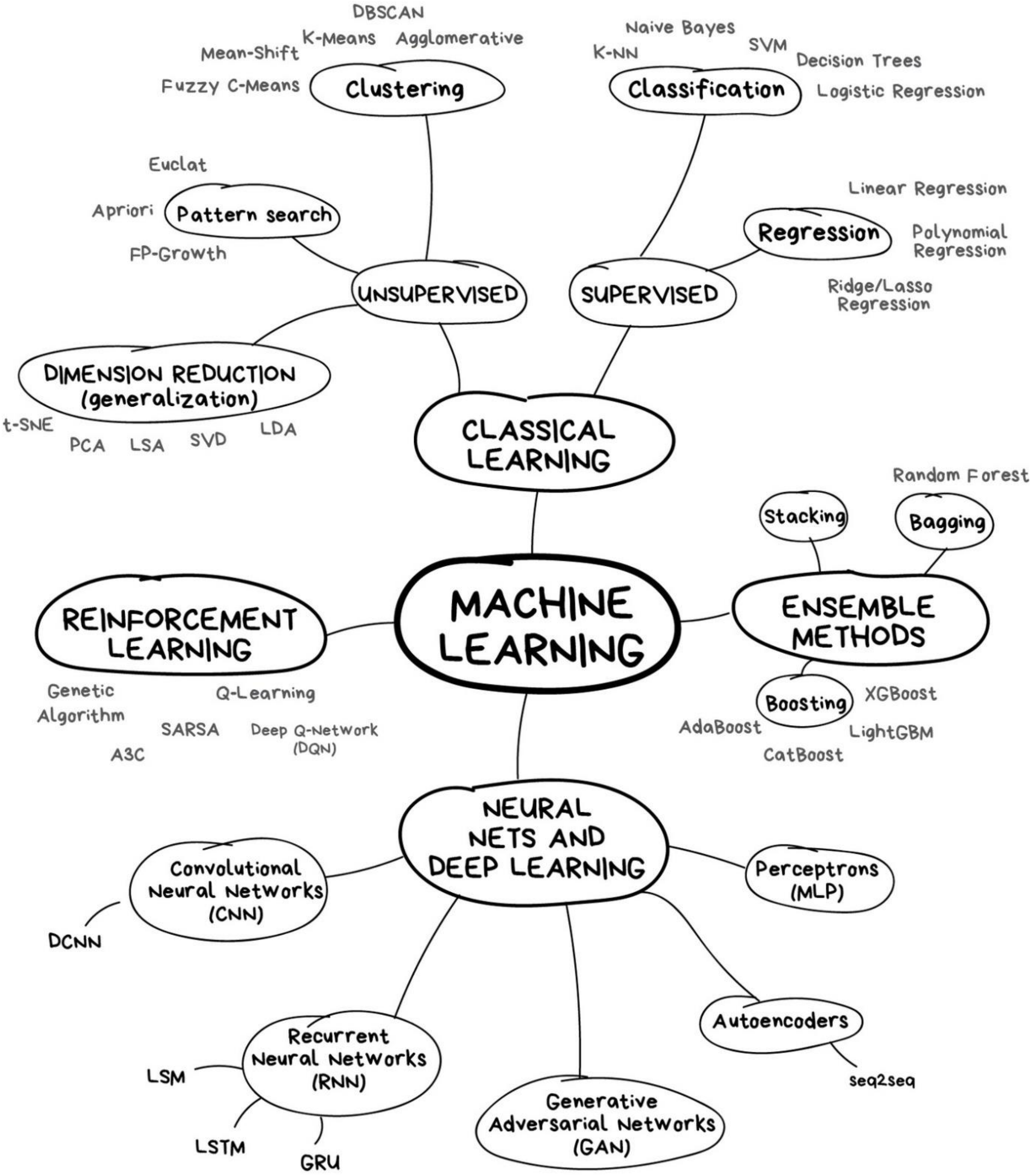

Figure 1. ML mind map. From "Machine Learning for Everyone," by V. Zubarev, n.d. (https://vas3k.com/blog/machine_learning/).