Alexander Gattis

11/9/2020

Advanced Artificial Intelligence

Individual Project Essay

Machine Learning and Implementation in Gaming

The concept of machine learning is a complex one, but also fascinating for those who have interest in the concept of machines and programs learning from people, or learning how to accomplish tasks that normally wouldn't be possible to be completed by anyone other than a human. In a sense, machine learning is the concept of teaching an AI how to accomplish a task through various methods, the most popular being through just repeated testing, or through supervised learning provided by datasets. Usually these datasets consist of video footage to give the AI something to follow and learn from. Or in even simpler terms, "Machine Learning is getting computers to program themselves. If programming is automation, then machine learning is automating the process of automation." (13)

Naturally, these concepts are very applicable to the video game industry. While they may not be as noticeable in most games, there are plenty of people who do AI experiments where they train an AI to play a video game, which usually takes machine learning to it's maximum potential. Today, you will be able to read about some of these projects and how they are important to the industry, as well as learn more about machine learning in general.

The goal of most machine learning algorithms is to provide the player with more immersion. How is that accomplished? Well, while one might think it would be to provide as much challenge through AI as possible, and while that is a valid method, some have other plans

for their algorithms. Surprisingly, a lot of immersion can come from playing with AI that acts as close to an actual person as possible, meaning the AI is prone to making mistakes, learning from them and playing as a normal person would play. While it's a bit difficult to actually implement this in a way that is optimized for people to play on general hardware, as "Training deep reinforcement learning agents complex behaviors in 3D virtual environments requires significant computational resources." (6) a lot of people have put this idea to the test to see what its capable of, by coding algorithms in certain games.

But how does this all work? Well, it all depends on what algorithm you use and how to go about using it. To put it into simpler terms, when you start testing a machine learning algorithm in a video game, usually it starts off very simple, if not unintelligent. It wont do very much and will need to be tested multiple times before it can actually learn to start doing things. Most learning algorithms are coded using a point or reward system. This means that when the AI dies or takes damage, it is seen as a punishment, and it will do anything in it's power to avoid that punishment from that point on. If the AI does something that benefits it, you can program the AI to see that as a reward, and it will go out of it's way to do that task when it can.

Testing an algorithm can be very time consuming, and can take hours, or even days or weeks, to get to a point where the average person would consider it to be able to play the game decently. But after it's tested for a really long time, it will start to bridge the gap between computer and human, and may even be able to accomplish feats that normally only a human could. Depending on what programming language and what game you're training on, the program used to test the neural network may be different. A popular one to use is Tensorflow, which is a program used to test games usually made in Python, and for very specific games,

you'll usually find that there are usually specific programs for someone to use to implement their algorithm.

There are many different algorithms to choose from, but ultimately it is up to the programmer to decide which is best for the specific instance of their research. And in some cases, they may even have to produce their own type of algorithm that hasn't been thought of or implemented into code yet. This can also be very time consuming and prove quite a challenge for most people. Some examples of algorithms include, but are not limited to; supervised learning, unsupervised, reinforcement, neural, etc. (11)

While some games may not necessarily implement this in a major way in the base game, it is however used for some things the average consumer may not even consider. For example, some machine learning is implemented into games to determine if someone is cheating or promoting bad behavior in the player base. (12) This includes but is not limited to; cheat software, quitting out of games mid-competition, toxic behavior in chat, etc. This type of machine learning algorithm can be used to promote better behavior and improve other players overall experience in multiplayer games.

One very big game in the machine learning community is none other than Minecraft, which is a very complex building and survival simulator game, which is a perfect test ground for a machine learning algorithm, because you can push it to the limit and see just what it can accomplish and learn from experimentation.

The machine learning and Minecraft community has done many things to experiment with this, such as creating the Minecraft supervised learning dataset MineRL, which has "over 60 million" (5) frames of data for people to test their algorithm with. This dataset can be used for

just about anything, however they do have AI programming competitions where people are supposed to create a machine learning AI that can find diamonds in the game as quickly as possible. And many examples and demonstrations exist online for people to view. Some have even taught their algorithm to learn how to do hardcore PVP (player versus player) combat by putting it in servers where its forced to adapt to players killing it over and over.

While being used for research and good intentions, sometimes the technology of machine learning can also be used for bad intentions, and this can be shown as well through games like Minecraft. This can be seen in how cheat software has been developed to help players get to good resources and cheat at the game by botting their account to do things for them. Of course, this is frowned upon in most communities and is pretty easily caught since most of the time it does not mimic human behavior, but is worth mentioning, nonetheless.

On another note, this kind of technology has also been used to teach agents and their algorithms to complete tasks. One such example is the experiment performed by Stephan Alaniz in their paper about using Deep Reinforcement and Monte Carlo Tree Search in order to “solve a block-placing task in Minecraft”. (7) This technology could definitely be implemented in video games in the future when technology gets better, such as having AI that builds randomly generated structures based upon datasets and examples given to it.

Another game that has had some machine learning implemented into is the popular fighting game franchise, Super Smash Bros. In particular the fan-favorite entry, Super Smash Bros. Melee which has it implemented due to its popular competitive scene. The implementation in question is an experiment involving the implementation of machine learning AI into a computer player in the game, with hopes that it would be able to compete, and possibly even beat top tier professional players. (3)

This type of algorithm is hard to implement into a fighting game for several reasons. The machine learning has to a significant amount of more work in order to determine it's decisions and this is because "From an RL standpoint, the SSBM environment poses several challenges - large and only partially observable state, complex transition dynamics, and delayed rewards." (4) The machine learning will take a lot longer to learn, because it can't see everything happening, and even more importantly it's harder for the algorithm to associate it's actions with a positive outcome due to the outcomes themselves being delayed due to the physics associated with the characters. Not to mention that there are over 20 characters each with different attributes that can affect how long it takes for the algorithms to adapt to them.

However, the algorithms themselves were able to overcome these issues. When placed in a competition that featured many professional players, it was able to hold its own and even beat a couple of top players when playing against them. There are even videos of their rounds together as proof available online to view. (14) But how did they go about testing these algorithms?

According to their research paper, they started training their algorithms by pitting it "against the in-game AI. After appropriate parameter tuning, both Q learners and actor-critics proved capable of defeating this AI at its highest difficulty setting." (4) The type of algorithms they used is considered in the industry as a Reinforced Learning algorithm, which means that the AI has to be tested over and over again, and each time it learns something new, with as little help from human supervision as possible. The other algorithm was a type of Q-learning, which is like reinforced learning, but that type of algorithm likes to take more unknown methods of achieving its goal.

After testing it against the in-game computer players, they noticed that the two algorithms had similar reward outputs to each other, but their methods of achieving those rewards were

different. While the RL agent tended to actually play more like a professional player, dashing around and using actual professional techniques and moves on the computer players, the Q-learning algorithm actually had its own special way of beating the AI. It actually started “tricking the in-game AI into killing itself” (4), which is a very impressive maneuver since according to the researchers it actually requires that the algorithm receiving “a small negative reward” (4) in the process, meaning it knows that despite taking damage it’s the most efficient way of winning.

They then began testing the AI against itself which improved its level of play even further. Finally, they took that current state of the algorithm and put it to the real test against several top level competitors in a smash tournament. The AI did very well against mid-level players at the competition, which is highly impressive for an AI to accomplish in such a weird game environment.

There are other times in the industry of machine learning where people make algorithms with the sole purpose of playing through a game as professionally as it possibly can. One of these algorithms is the MarI/O algorithm, which is an algorithm for Super Mario World which has the purpose of making it to the end of a level in the game. (8)

The type of algorithm used was that of neural network node evolution. Specifically, an algorithm coined as the NEAT algorithm by those who made it and use it. The basics of this algorithm is that the AI is able to see partial bits of the level and associate those parts with neurons, which then connect to inputs for buttons. The algorithm is heavily unsupervised, meaning it requires a lot of testing and the AI starts off with no knowledge whatsoever but slowly and gradually starts to learn after failing over and over.

By the end of testing in the experiment, the AI was able to accomplish great feats and beat the level without taking any damage, something that most humans would not be able to accomplish. The same programmer went on to make another project which ironically enough, also involves Mario. This project was called MarIQ.

MarIQ is an algorithm created by the previous programmer with the intentions of teaching an AI to play through levels of Super Mario Kart for the Super Nintendo Entertainment System. (10) The algorithm uses the algorithm of Q-learning, which is the same type of algorithm used in the previous project involving Super Smash Bros.

An interesting thing about this specific algorithm project is that there is actually a UI integrated into the software in order to make it easier for people to understand how it functions. The demonstration video is a very good dive into the basics of Deep Q-Learning and the concepts of machine learning and testing in general. The algorithm starts off knowing nothing as usual, but as it starts pressing random inputs, it will either receive a negative reward or positive reward, which will contribute to its overall score.

Obviously, the AI's main goal is to keep its score as positive as possible, therefore it will do anything in its power to make sure it gets as many positive rewards as possible. In this case, positive scores are linked to whether the kart, or in this case the AI, is making forward progress through the level. Getting back to the UI of the project, the algorithm contains a series of arrows which represent the AI's confidence in it's abilities to attain more positive rewards. This is a good way to represent how an AI is thinking to the average person.

This algorithm also incorporates a system that the programmer coins as the "Tryhard" system. This is a system that allows the AI to occasionally switch things up by determining how

little or how much it needs to rely on it's neural network, based upon the percentage provided to it. This allows it to possibly unlock new techniques or discover new ways of attaining it's positive rewards even faster.

An unintentional discovery of this project was an example of generalization. This is the ability for a machine learning AI to be able to be dropped into a brand new environment with new obstacles and hazards, but still be able to hold it's own as it would the level it was trained on. It's a good way to prove that your AI has been tested successfully, since it means that you could in theory implement it in any level and get interesting results out of it.

In conclusion, the concept and development of machine learning has accomplished incredible feats and there is no telling where it may go next. However, hardware is a very big issue when it comes to implementing it. Unless you have a lot of spare processors or a supercomputer, it'll be hard to actually perform and implement experiments like this on your own. While the games industry is not quite ready to have it implemented fully into consumer-grade products, the industry itself is growing so quickly that there is no telling what the future for video game AI holds.

Bibliography:

(1) Geisler, B. (n.d.). *Integrated Machine Learning For Behavior Modeling in Video Games* (pp. 1-9, Rep.). https://www.aaai.org/Papers/Workshops/2004/WS-04-04/WS04-04-012.pdf

(2) Bowling, M. (2006). *Machine learning and games* (pp. 1-5, Rep.). Springer Science + Business Media, LLC.

(3) Gershgorn, D. (2017, February 23). A Super Smash Bros-playing AI has taught itself how to stomp professional players. Retrieved October 17, 2020, from https://qz.com/917221/a-super-smash-bros-playing-ai-has-taught-itself-how-to-stomp-professional-players/

(4) Firoiu, V. (2007). *Beating the World's Best at Super Smash Bros. Melee with Deep Reinforcement Learning* (pp. 1-7, Rep.). https://arxiv.org/pdf/1702.06230.pdf

(5) https://minerl.io/ (A popular Minecraft Neural Network dataset)

(6) Frazier, S., & Riedl, M. (n.d.). *Improving Deep Reinforcement Learning in Minecraft with Action Advice* (pp. 1-7, Rep.). Georgia Institute of Technology.

(7) Alaniz, S. (n.d.). *Deep Reinforcement Learning with Model Learning and Monte Carlo Tree Search in Minecraft* (pp. 1-5, Rep.). Berlin, Germany: Department of Electrical Engineering and Computer Science Technische Universitat Berlin. https://arxiv.org/pdf/1803.08456.pdf

(8) https://pastebin.com/ZZmSNaHX (MarI/O Super Mario World Neural Network)

(9) Stannely, K. O. (n.d.). *Evolving Neural Networks through Augmenting Topologies* (pp. 1-30, Rep.). The MIT Press Journals. http://nn.cs.utexas.edu/downloads/papers/stanley.ec02.pdf ("NEAT" algorithm used for MarI/O)

(10) SethBling (Director). (2019, June 29). *MarIQ -- Q-Learning Neural Network for Mario Kart* [Video file]. Retrieved from https://www.youtube.com/watch?v=Tnu4O_xEmVk (MarIQ Experiment)

(11) Ray, S. (2017, September 9). Commonly used Machine Learning Algorithms (with Python and R Codes). Retrieved from https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/

(12) Ray, S. (2017, September 9). Commonly used Machine Learning Algorithms (with Python and R Codes). Retrieved from https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/ (Machine Learning Cheat Catching)

(13) Brownlee, J. (2020, August 15). Basic Concepts in Machine Learning. Retrieved from https://machinelearningmastery.com/basic-concepts-in-machine-learning/ (Machine Learning Basic Concepts)

(14) https://www.youtube.com/watch?v=dXJUlqBsZtE&feature=emb_title (Smash AI example)