Learning Algorithms

Braylin Jones-Mcmullen

AI learning algorithms, known as machine learning, are a set of rules that help a system learn from the data that is given to it. AI development and research is a very specialized field and is a subfield of computer science. Programmers take this knowledge and apply it to many applications in our lives like spam folders, auto correction and predictive text in word documents or on custom keyboards, and most notably, video games. Artificial intelligence can range from the directions four sprites go in while chasing pacman, to bringing complete life to the city of Los Santos in Grand Theft Auto V and everything in between.

There are two different types of learning, online and offline learning. Any form of learning done during gameplay is considered online learning. While learning online, the computer dynamically analyzes the player’s moves and can better predict and anticipate what the player will do next. This can be beneficial for fighting games like smash bros. or music training games like rocksmith. But the benefits of online learning are its negatives as well. Online learning produces too many problems with predictability since the game is constantly changing based on the player’s decisions. This also makes it hard to find bugs and problems since many instances will rarely be replicated.

With that knowledge we can deduce that majority of learning is done offline in between levels or, more often, at the development studio before the game leaves. When learning in between levels, characters aren’t learning as they are acting. Sadly this does not alleviate us from similar weaknesses from online learning. A bug can still be found during load screens if a player triggers a certain attribute. These bugs may not be found until the next level. Developers must make sure that bugs and problems can be replicated without replaying the game several times. No game or system implements a learning system the same way and each type of algorithm has its own set of strengths and weaknesses.

A familiar and easy to read form of learning is the decision tree algorithm. This form maps out the inputs provided to it and updates a given piece of data accordingly. As the name entails, decision trees can explicitly represent decisions and the inputs provided can alter decisions. When it comes to learning algorithms, decisions trees break down to separate parts.

One of the most notable is the ID3 or the inductive decision tree algorithm 3. It uses recursion where it starts off with a single node and then replaces it with decisions until the tree is created. The data for this system is called attributes and it uses these attributes to build the decision tree. A huge benefit of this set up is that it builds the fastest and shortest tree. This is beneficial when the developer has a set base of attributes that will be used and tested and this is perfect for offline use.

The weakness with ID3s is when it is online. When online, new examples will generate while the game is running. The ID3 algorithm is able to adjust and re-run all the information given to it to create a new tree but it is not efficient when dealing with large databases of examples and can become very time consuming.

Another form of learning algorithm is reinforcement learning. This technique provides learning based on experience. It is best used on a single character and can help it better choose actions over time. Its benefits greatly by using a reward value system where a given decision can be considered a good choice and lead to goal. The algorithm will look at the steps it took to get to that goal during multiple instances and learn about what is considered a good choice or not. This is also the same case for negative rewards. Getting them will help the character learn about what is bad also.

The biggest weakness of reinforcement learning is its very small acceptance in the game design community. Some companies have invested in more research of this topic because of its promising techniques but it just isn’t practical for many applications. And the algorithm requires the game to be represented as a set of states linked by actions. This can be very complex, time consuming, and memory hungry if not controlled the right way.

This is just a small look at some of the strengths and weaknesses of learning algorithms and a deeper look at a couple forms of learning. Each proves a very strong point but also has a strong weakness that undercuts it. I have not found a case for an algorithm that has no or little weakness but this isn’t the case for anything in programming. Although learning is roughly used in mainstream, it should definitely be looked into especially with where the power of games is moving too. Microsoft boasts the power of “the cloud” for this reason.