INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 3)
Available online at: www.ijariit.com

Artificial neural networks implementation in digital games

Prem Motgi
motgi.prem@gmail.com
Sinhgad College of Engineering, Pune, Maharashtra

ABSTRACT

An Artificial Neural Network is a framework that tries in different degrees to copy a human cerebrum with a specific end goal to perform assignments that other PC frameworks are generally not fit to handle. As the field of AI and learning frameworks utilizing counterfeit neural systems is being created and extended, it is inescapable that its utilization in gaming will be investigated thoroughly. In the Digital Gaming among these years, the games can be cracked easily because there is not a much tougher opponent in games and the games can be predicted as well so that the user is easily able to know about upcoming moves in games. So Need of ANN here is, its ability to learn and adaptation of many different tasks and make complex predictions. ANN in games learns the way in which the player is playing the game and according to that, it makes its new move so that the game becomes much tougher and enriches the experience of playing games. In this paper reviews about the implementation of ANN and its algorithms for Board Games, Modern Games and digital video games as well.

Keywords: *ANN, Digital Games, Neural Systems, Prediction, AI Prediction, AI.*

1. INTRODUCTION

An **Artificial Neural Network** is a system that tries in various degrees to emulate a human brain in order to perform tasks that other computer systems are usually not fit to handle. As an opponent AI has a huge impact on a game's enjoyability, the ability to such an opponent to learn and get better over time is intriguing. Usually, the actions of non-player-characters (NPCs) in games are either completely pre-determined or rule-based and dependent on different conditions being met. This is why the idea of a learning and evolving NPC, by means of an ANN, is very interesting. Different kinds of ANNs have been used, to create life-like and intelligent opponents for classic board games such as Checkers and Othello, as well as for modern PC games such as Unreal Tournament. ANNs have also been implemented for other tasks, such as detecting cheaters in online games and for dynamic gesture recognition in dance games and in most famous games like Nero. ANNs have additionally been enforced for different tasks, like detecting cheaters in online games and for dynamic gesture recognition in dance games. Online learning may be a way tougher prospect because it's a real-time process and many of the normally used algorithms for learning are so not suitable. Instead, these algorithms should be adapted for period dynamic processes zero. Real-time strategy (RTS) games are a particular candidate for online learning algorithms and a few interesting approaches are being developed. In some situations, a combination of each offline learning and on-line adaptation is that the most acceptable approach. These aspects of the implementation of learning technologies into games are inherent to the use of neural networks in games.

2. LITERATURE REVIEW

In gaming, the computer-controlled opponent behavior is typically rule-based mostly and dependent on specific conditions and therefore, be predictable to a definite degree. Because the field of AI and learning systems using artificial neural networks is being developed and expanded, it's inevitable that its use in gaming is explored completely. This short survey appearance at the tries of using artificial neural networks for opponents in board games and trendy computer games, yet as alternative uses in gambling throughout the last twenty years. The ability to evolve and learn presents nice possibilities for the use of ANNs as AI in games. Within the earlier stages throughout the late Eighties and early Nineties, it was discovered that ANNs might be wont to produce adaptive and creative AIs that weren't rule-based and failed to require expert knowledge or vital a priori knowledge to train.

3. HISTORY

Artificial Neural Networks (ANNs) have been in wide use since at least the 1980s for, among other things, complex modeling and various recognition, prediction and filtering tasks. Their ability to learn and evolve has made them attractive to many different fields of research and innovation, including gaming. As an opponent AI has a huge impact on a game's enjoyability, the ability for such an opponent to learn and get better over time is intriguing.

Over the last 20 years, computer scientists, engineers, and programmers have experimented with the possibilities of the ANN in games. Different kinds of ANNs have been used, to create life-like and intelligent opponents for classic board games such as Checkers and Othello, as well as for modern PC games such as Unreal Tournament. ANNs have also been implemented for other tasks, such as detecting cheaters in online games and for dynamic gesture recognition in dance games. This short survey attempts to summarize the most

significant attempts and their results, as well as provide a brief look at the future possibilities of ANNs in gaming.

In an entirely logical sense, artificial neural systems are mathematical models for non-linear functions. Trained ANNs are utilized to give craved yield from an arrangement of input data parameters without the requirement for a precise capacity or model for the issue. ANNs can be seen as a simple copy of their biological partners. All ANNs just as the brain, consist of neurons, also called nodes in ANNs, but while the human brain encases around 100 billion neurons with thousands of connections between them, the typical ANNs are on a much smaller scale. This paper will quickly talk about the fundamental components of ANNs.

4. ANN IN DIGITAL GAMES

Most writings on neural networks begin with a biological description of neurons in our brains as an image for a way artificial neural networks function. Specifically, you'll consider a neural network as a function more approximate. The input to the network represents independent variables, whereas the output represents the dependent variable or variables. The network itself is then a operate giving one distinctive set of output for the given input. For games, neural networks supply some key benefits over additional traditional AI techniques. First, employing a neural network could permit game developers to make ease in the coding of complicated state machines or rules-based systems by delegating key decision-making processes to 1 or more trained neural networks. Second, neural networks supply the potential for the game's AI to adapt as the game is played. This is often a rather intriguing possibility and is a highly regarded subject within the game AI community at this time.

ANN Techniques and Algorithms used in Digital Gaming:

1. Supervised and Unsupervised Learning
2. Reinforcement Learning and Back Propagation
3. Evolutionary Learning Algorithm
4. NEAT and rtNEAT

4.1. Supervised and Unsupervised Learning

With supervised learning, the network is offered with input data and therefore the correct answer i.e. what output is likely to receive only if input data. The input data is often propagated forward through the network till activation reaches the output neurons. The comparison of the answer that the network has calculated therewith that it tends to needed to get. If the answers agree, no modification to the network; if, however, the solution that the network is giving is completely different from that that is needed then change the weights to make sure that the network is more likely to give the correct answer in future if it's again given with an equivalent (or similar) input data. This weight adjustment scheme is understood as supervised learning or learning with a tutor.

With unsupervised learning there's no external teacher and learning is usually primarily based solely on data that's native to every neuron. Typically this can be additionally often said as self-organization, within the sense that the network self-organizes in response to knowledge presented to the network and detects the emerging collective properties among the data. Unsupervised neural strategies are usually utilized in an exploratory manner and have a tendency to use statistical relationships between data variables so as to establish an understanding of the character of the data. In contrast to supervised learning, we have a tendency to don't grasp the answers before we start training.

4.2. Reinforcement Learning and Back Propagation

Reinforcement learning relates to increasing a numerical reward signal through a form of trial-and-error search. so as to find out the network isn't told that actions to take however instead should discover that actions yield the foremost reward by ANN attempting them – if an action has been successful then the weights are altered to strengthen that behavior otherwise that action is discouraged within the modification of the weights. Reinforcement learning is completely different from supervised learning in that with supervised ways, learning is from examples provided by some knowledgeable external supervisor.

Backpropagation needs specific information on the required output of the ANN. It calculates the error between the required output for the network and therefore the actual output of the network. Every node gets a complete error from all of its outputs, and so adjusts the weights to its input connections consequently. This method is propagated backward from the output layer all the manner through the input layer of the network, giving it its name "back propagation". An additional elaborated description of backpropagation is explained in [1] together with alterations of the formula to enhance performance, which is additionally mentioned in [2].

4.3. Evolutionary Learning Algorithm

Evolutionary Algorithms (EAs) and Genetic Programming (GP) are used to optimize parameters to find the best possible solution to a problem [3]. To use EAs, a population of possible solutions as individuals must be acquired. In the case of ANN training, an individual would be an ANN, with its associated weights and architecture. Individuals are evaluated with a fitness function, depicting how well they solve the problem. Then, the fittest solutions generate "offspring" which are solutions whose parameters are copies or combinations of the parent solutions', usually with a degree of random mutation. This way, the genetic algorithm strives to find the best solution by letting the population of individual solutions evolve over a multitude of generations. In the case of ANN training, this would result in finding an optimal set of weights to be used in the neural network, or if specified, even the optimal structure.

4.4. NEAT and rtNEAT

NEAT stands for Neuro Evolution Argumenting Topology.
Video games have several properties that pose significant challenges to traditional RL:

1) Large state/action space
2) Diverse behaviors
3) Consistent individual behaviors
4) Fast adaptation
5) The memory of past states

All these properties are fully satisfied by NEAT and not by Reinforcement algorithm. Thus, NE is a good match for video games. The current challenge is to achieve evolution in *real time*, as the game is played. If agents could be evolved in a smooth cycle of replacement, the player could interact with evolution during the game and the many benefits of NE would be available to the video gaming community. This paper introduces such a real-time NE technique, rtNEAT, which is applied to the NERO multi-agent continuous-state video game. In NERO, agents must master both motor control and higher-level strategy to win the game. The player acts as a trainer,

teaching a team of robots the skills they need to survive. The next section reviews the NEAT neuroevolution method, and how it can be enhanced to produce rtNEAT.

All these properties are absolutely satisfied by NEAT and not by Reinforcement algorithmic rule. Thus, NE could be a smart match for video games. This challenge is to realize evolution in real time because the game is played. If agents might be evolved during a smooth cycle of replacement, the player might interact with evolution throughout the game and also the several advantages of NE would be accessible to the video gaming community. There is such a period of time NE technique, rtNEAT, which is applied to the Nero multi-agent continuous-state game. In NERO, agents should master each motor control and higher-level strategy to win the game. The player acts as a trainer, teaching a team of robots the skills they need to survive. Ensuing section reviews the NEAT neuroevolution methodology and the way it is often increased to produce rtNEAT.

rtNEAT performs the following operations:
1. Remove the agent with the worst *adjusted* fitness from the population assuming one has been alive sufficiently long so that it has been properly evaluated.
2. Re-estimate F_bar for all species.
3. Choose a parent species to create the new offspring
4. Adjust compatibility threshold Ct dynamically and *reassign* all agents to species.
5. Place the new agent in the world.

5. IMPLEMENTATION IN DIFFERENT GAMES

ANN is used in developing an expert opponent in various types of games such as:

1. Board Games
2. Moderate Modern Games
3. Racing Games
4. Real-Time Games

5.1. Board Games

Othello

A prominent usage of ANNs for AI is for the round of Othello, otherwise called Reversi. One execution [4] was finished by David E. Moriarty and Risto Miikkulainen in 1995. Moriarty and Miikkulainen created and trained an ANN capable of discovering complex strategies and playing on par with an experienced human. The ANNs which got the 8x8 board setup and delivered the following best position was advanced utilizing an EA. Moriarty and Miikkulainen utilized a populace of 50 ANNs spoke to utilizing the Marker-Based Encoding plan, clarified in the subtle element in [4], taking into account advancement of weights and design, bringing about dynamic architectures for the ANNs.Other board games-tic tac toe and five in a row.

Fig. 1: Screenshot of game Othello

5.2. Moderate Modern Games

After ANNs were used successfully for board game AI opponents, it was a natural consequence that ANNs would then be applied to modern computer games. This can be a bigger challenge than implementing AI for board games, as the rules may not be as clear and it may be hard to determine what behavior is suitable.

The AI opponent ("bots") in UT2004 have an arrangement of conceivable practices that are adjusted relying upon state characteristics, for example, if the bot sees an adversary and what number of wellbeing focuses it has. These standards are customized physically. This test attempted to imitate the practices of the purported "Seeker Bot" with SGNNs and FALCON by both logged off preparing and testing, and online testing. In the offline training, the inner states, outer states and conduct examples of the Hunter Bot were utilized as preparing and testing information for both the two SGNN bots (one without the pruning) and the FALCON bot.

Among 8000 test and training samples The FALCON bot, however, had basically the same capabilities as the Hunter bot and tied with it in games. Thus it was shown that while both SGNN and FALCON could be used to create bots in UT2004, Falcon was the most suitable.

Fig. 2: Screenshot from the Game UT2004

5.3. Racing Games

ANNs have additionally been utilized to control autos in racing games. H. Tang, C. H. Tan, K. C. Tan and A. Tay looked at the potential outcomes of using so as to make AI for hustling amusements neural systems or Behavior Based Artificial Intelligence (BBAI) [5]. The dashing test system utilized required two players to drive through waypoints in a chose arrangement for a set measure of time. There are just two waypoints unmistakable whenever, where one is dynamic and one is definitely not. The player that had driven through most waypoints when the time was up had won. The neural network was trained with an evolutionary algorithm, running for 200 generations with a population size of 100. The individuals were evaluated by comparison with the Heuristic Sensible Controller, which was one of 3 basic heuristic controllers, similarly to how the Hunter Bot was used in the UT2004 experiment.

Fig. 3: Screenshot from the game "Mac O Rally 2"

5.4. Real-Time Games

NERO

NERO is representative of a new genre that is only possible through machine learning. The idea is to put the player in the role of a *trainer* or a *drill instructor* who teaches a team of agents by designing a curriculum.

In NERO, the learning agents are simulated robots, and the goal is to train a team of robots for military combat. The robots begin the game with no skills and only the ability to learn. In order to prepare for combat, the player must design a sequence of training exercises and goals specified with a set of sliders. Ideally, the exercises are increasingly difficult so that the team can begin by learning a foundation of basic skills and then gradually building on them. When the player is satisfied that the team is prepared, the team is deployed in a battle against another team trained by another player (possibly on the internet), making for a captivating and exciting culmination of training. The challenge is to anticipate the kinds of skills that might be necessary for battle and build training exercises to hone those skills. The next two sections explain how the agents are trained in NERO and how they fight an opposing team in battle.

rtNEAT algorithms are mostly used for the real-time games rather than using the NEAT algorithm. NERO game is also implemented with rtNEAT.

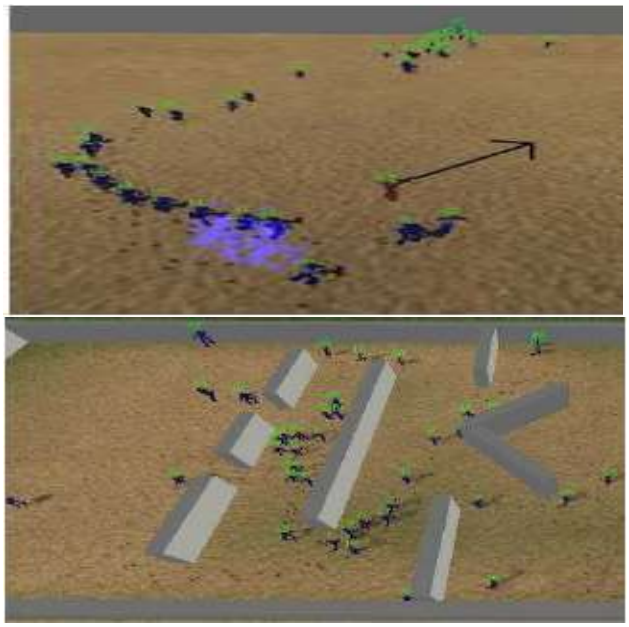

Fig. 4: Screenshot from the game NERO showing that the turrets are getting trained and chooses another path for navigating the maze.

6. BOTTLENECKS OF ANN

As the ANN in Digital Gaming is still in the development phase there are some disadvantages that are needed to overcome and that is:

1) Troubleshooting the implementation of the algorithm is tricky.
2) ANN needs a large diversity of training (from games) for the operation.
3) Slow Convergence speed.
4) Less Generalizing performance.

7. CONCLUSION

These AIs were initially enforced on board games like Reversi or Tic Tac Toe, given their easy rules and playing environment. Through completely different experiments evolving ANNs through evolutionary algorithms or alternative training strategies, AIs that compete at levels rivaling high human players and AIs based on expert knowledge were developed on many fronts. With modern-day computational power, variations of evolutionary algorithms and implementation of fuzzy ANNs have been used to measure ANNs as AI for modern games. In board games, the AI typically has time to work out numerous completely different moves and outcomes and select the foremost favorable action consequently. In computer games compete in the period, there are much more durable time constraints. Luckily, the character of the ANN makes it terribly appropriate for these types of period tasks also, as has been demonstrated by its use in first person shooters and racing simulators. ANNs have also shown applications in predicting player expertise in games, the outcome of sports matches, and helping diagnose psychological feature disorders. Gaming after training (800 samples) with MLP and FTFLN network had the best results. It could correctly detect 21.14% of illegal moves and produced 0.98% false positives. The FTLFN while being able to correctly detect 59.27% of illegal moves, unfortunately, generated a 4.4% for false positives and was deemed inferior because of this. ANNs gives significant advantages in cheat detection and good results in accuracy.

8. REFERENCES

[1] Ng, S.C, Leungh, S.H, and Luk, A. “the generalized back-propagation algorithm with convergence analysis”. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, 1999 (ISCAS ’99), Volume 5, May 30 – June 2, 1999, p. 612-615.

[2] Lee, H. M, Huang, T.C., and Chen, C.M., “Learning Efficiency Improvement of Back-Propagation Algorithm by Error Saturation Prevention Method”. International Joint Conference on Neural Networks 1999 (IJCNN ’99), Volume 3, 10-16 July 1999, p. 1737 – 1742.

[3] Koza, J.R. “Survey of Genetic Algorithms and Genetic Programming”, WESCON/'95. Conference record. 'Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies', 1995

[4] Moriarty, David E, Miikkulainen, Risto. “Discovering Complex Othello Strategies Through Evolutionary Neural Networks”, Connection Science, Volume 7, 1995, p. 195-209.

[5] Tang, H., Tan, C. H., Tan K. C., and Tay, A. "Neural Network versus Behavior-Based Approach in Simulated Car Racing Game". IEEE Workshop on Evolving and Self- Developing Intelligent Systems, 2009. ESDIS '09, March 30 2009-April 2 2009, p. 58 – 65.

[6] Bellemare, M. G., Veness, J. & Bowling, M. Investigating contingency awareness using Atari 2600 games. Proc. Conf. AAAI. Artif. Intell. 864–871 (2012).

[7] Lange, S. & Riedmiller, M. Deep auto-encoder neural networks in reinforcement learning. Proc. Int. Jt. Conf. Neural. Netw. 1–8 (2010)

[8] O’Neill, J., Pleydell-Bouverie, B., Dupret, D. & Csicsvari, J. Play it again: reactivation of waking experience and memory. Trends Neurosci. 33, 220–229 (2010).

