Neural Network
Based AI for mobile games

Sheetanshu
New Delhi, India
Member of Digital Monozukri Team
www.dm.amidstsky.com
sheetanshu.sinha@gmail.com

Prof. Dr. Aynur Ünal
Founder, Digital Monozukuri, Palo Alto, CA, USA
aynurunal@stanfordalumni.org

***Abstract* :-The rise in technology had driven us to the emergent of many Computational Intelligence techniques Neural Networks being prominent of the one. Neural Network activity and machine learning have received a very dynamic and responsive results in almost every field requiring Computational Intelligence as a substitute. The various AI techniques in present digital games are behavior based approach, control theory, Extrusive data mining, heuristics search based methodologies. Intelligent control with expert systems, fuzzy logic and neural networks are recently being explored as an effective substitute to these methods. The AI complexity in present games have been a very much selling point by the developers. High performance and effective AI for a good gameplay within envoirment restrictions are very fascinating and challenging subjects of research and development.**

keywords: Artificial Neural Networks; game AI; simulation; mobile games Ar.

I. INTRODUCTION

The use AI in games is to simulate human like intelligence, or intelligent behaviors for your Non-Playable characters, as well as envoirment interactivity and restrictions for good realistic gameplay experience. AI have always been the power house and heart of any gameplay and the simultaneous implementation is a very challenging and interesting area for game programmers and gaming industries. Our present approaches in solving problem statements of Game AI haven't much progressed from the traditional and brute force techniques developed in the late 70s or so. The present AI behavior in games relies heavily on some form of search over the possible outcomes from the intelligent subset with data mining and analytics, for the Optimal efficiency, rather than something what we can term as an intelligent behavior. Developing AI for present games have become a complicated endeavor since today most of the games use tight computational constraints, with high end graphics, dynamic interactive envoirment along with physics simulation, which leave a very less amount time of time in the CPU cycle for computing the AI. Many major game genres requires to update the AI many times a second, leaving the AI to manage large number of agents, in potentially a very less processing time.

The ANN technique depicted here is both computationally inexpensive as well as, easy implement, which will address the constraint found in most game development.

II. LITERATURE SURVEY

Disparate amount of comprehensive articles, journals and conference papers can be found describing various AI techniques and methods applicable in game programming and gamification. Different methodologies and algorithms are described in these various articles and papers to deduce an effective AI behavior. Some uses behavior based selection techniques to perform actions and response in the game world, while others use tightly coupled data mining and heuristic search approaches. Nevertheless some literatures that are indirectly related to this thesis are used as a basis in this discussion.

Traditionally game AI parameters are nothing but user's actions and coupled responses, and AI computation is n brute force outcome of these parameters in account with heuristic search from and intelligent outcome subset. Yet as stated earlier this method need to be computationally inexpensive as well as efficient for the short computational time it's getting from the CPU cycle. This methodology can greatly increase the accuracy as well as well as provide an in expensive method for the same.

III. METHOD AND TOOLS

A. TOOLS AND DESIGN

We will be making use of Unity 3D a very powerful game engine, which is presently an extremely popular choice among developers for game design and building games, with approx. 1.2 million registered developers and 6 million worldwide developers. 53% of mobile game developers now use unity 3D today.

We will setup a normal scene, with two bots, surrounded by enemies, we will set up the plain along, add the appropriate physics, colliders and program the collision detection, with bullets. We will also set up the lights, Directional lights being the choice here, along with the skyboxes, and set up a re spawn code attached to an empty game object at the re spawn point. The will be attached with the mouse scroller to prompt view from a third person perspective.

We will be using a corresponding neural network library plugin, designed especially for unity 3D. The procedure will start with making up a triangular set values (0,1), the first being the normalized health of the bot, second being the normalized number of zombies near the bot(0.1 stating 1 bot, 0.2 2 bots, 0.3 3 bots and so on), the last number will be the trigger where we will programmatically check, in accordance with the following table.

<0.3	**Do Nothing**
>0.3 &&<0.6	**Run away**
>0.6	**Shoot**

Table No.1.1 Table showing constraints

The training set will be formulated in a text file as well as the output will be recorded in the same format too. This design keeps in accordance two input parameters and one output parameters, hence the first layer will have 2 neurons and the last layer one, and will be using 7 neurons in the hidden layer. A preceptor based training will create the AI based on the constraints, provided above ant the output will be in the form of, array of outputs.

IV. SIMULATION AND ANALYSIS

The simulation will be creating the AI based on the study of triangular set provided below:-

1	**0**	**0**
1	**0.1**	**1**
1	**0.2**	**1**
1	**0.25**	**1**
1	**0.4**	**0.4**
1	**0.5**	**0.4**
1	**0.6**	**0.4**
0.7	**0**	**0**
0.7	**0.05**	**1**
0.7	**0.1**	**1**
0.7	**0.2**	**0.4**
0.7	**0.3**	**0.4**
0.5	**0**	**0**
0.5	**0.05**	**1**
0.5	**0.15**	**0.4**
0.5	**0.2**	**0.4**
0.3	**0**	**0**
0.3	**0.05**	**1**
0.3	**0.1**	**0.4**
0.3	**0.2**	**0.4**
0.15	**0**	**0**
0.15	**0.05**	**0.4**
0.15	**0.2**	**0.4**

Table No.1.2 Table showing the test data of Induction motors

Results

Network structure

2	**7**	**7**	**1**

Input Layer	Hidden Layer	Hidden Layer	Output Layer
1.708223	2.559051	-4.650617	-5.292803	9.301042
-5.825083	1.577219	1.001748	-0.6150293	-2.110187
5.588218	6.691418	25.45093	-53.56604	

Table 1.3 Layer 1

0.2686256	-0.1570961	-1.61739	0.5627522	-0.9232384	-0.4057407	-0.3969513
0.04747073	-0.5862995	-0.5559512	3.702804	-0.161915	-0.2987756	-0.8871361
1.039362	-0.3834676	1.20786	3.964347	-0.299512	0.6759934	-1.064715
-0.06645595	-0.07084517	-2.785209	-4.48268	0.4061259	1.044816	-0.6969025
-0.487059	-1.064554	-2.700357	1.930266	-1.734056	-0.8607444	-0.961219
-0.5452474	1.124389	-2.056758	-9.20648	-0.3743652	0.09391341	1.431719
-1.158745	2.326056	11.05472	-8.858867	5.034259	0.03634342	3.4497

Table 1.4 Layer 2

-0.03449466	-0.7364625	-3.492471	4.507268	-0.6983762	0.1435606	-1.059121

Table1.5 Layer 3

V. FUTURE SCOPE

- ANN based AI in games can be an effective solution to obtain a computationally inexpensive as well as efficient AI for modern games where more emphasis is on the game world, graphics and physics system.
- Many networks like hopefield pattern recognition network, Kohonen self-organizing network applications are yet to be explored.

CONCLUSION

The method discussed the application of Artificial Neural Networks in building a computationally cheap in expensive game AI applicable to all types of commercial as well as mobile games. The proposed method very small computational resource of apporax. 1% to 5%. The ability to learn present great possibilities in game application as well . After satisfactory results from the simulations performed the code segmented was successfully integrated and implemented in small mobile games and the average error is 0.0499, which is within the permissible limits. This had shown very promising results for the development of new and less predictive AI. With the new types evolutionary algorithms and strong computational powers, this field would be very much worth spending for game developers, building AI.

ACKNOWLEDGEMENT

We would specially like to thank Dr. Aynur Unal (Stanford Alumni) for her immense contribution to this paper. She has been into the research field since the decades and has published many technical papers. This paper has been completed under the supervisional guidance of Dr. Aynur Unal.

REFERENCES:

[1] Zadeh, L. A. "Fuzzy Algorithms", Information and Control, Vol. 12, pp.94-102, 1968

[2] Aynur Unal "Detecting Ball Bearing Defects Using Artificial Neural Networks," Jrnl of Acoust. Soc. Am. Suppl.1, Vol. 88, pp.20, November 1990.

[3] Aynur Unal, "Detecting Ball Bearing Defects Via Artificial Neural Networks," WNN 91 Conference, February 11-13, 1991, Auburn, Alabama.

[4] Aynur Unal, "Diagnostics of Ball Bearings Using Neural Nets," Active Control of Sound and Vibration, April 15-17, 1991, Blacksburg, Virginia.

[5] Aynur Unal, "Diagnostics in Predictive maintenance Using Neural Networks and Acoustic Signatures," First International Conference on Acoustic Emission in Manufactuirng, June 3-6, 1991, University of California, Berkeley.

[6] Aynur Unal "Detecting Ball Bearing Defects Using Artificial Neural Networks," Jrnl of Acoust. Soc. Am. Suppl.1, Vol. 88, pp.20, November 1990.

[7] Aynur Unal, "Detecting Ball Bearing Defects Via Artificial Neural Networks," WNN 91 Conference, February 11-13, 1991, Auburn, Alabama.

[8] Aynur Unal, "Diagnostics of Ball Bearings Using Neural Nets," Active Control of Sound and Vibration, April 15-17, 1991, Blacksburg, Virginia.

[9] Aynur Unal, "Diagnostics in Predictive maintenance Using Neural Networks and Acoustic Signatures," First International Conference on Acoustic Emission in Manufactuirng, June 3-6, 1991, University of California, Berkeley.

[10] 2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings. NAFIPS-FLINT 2002 (Cat. No. 02TH8622)

[11] Aynur Unal" Acoustic emissions as diagnostic tools for complex systems." J. Acoust. Soc. Am. 90, 2323 (1991)

[12] Aynur Unal "Vibration signature analysis using artificial neural networks "J. Acoust. Soc. Am. 88, S20 (1990)

[13] Ünal, A., "Snow Cap" Patent for electronics manufacturing e-markets, E2open, Feb. 2001.

[14] Ünal, A., "Generalized e-hubs: A Stack of Pancakes" Patent for electronics manufacturing e-markets, E2open, January 2001.

[15] Ünal, A., "Integrating Different Processes of Electronics Manufacturing as Generalized e-hubs: Revolving Doors" Patent for electronics manufacturing e-markets, E2open, January 2001.

[16] Condition monitoring and diagnostics of systems and its relationship to some aspects of artificial intelligence.

[17] Beal C, Beck J, Westbrook D, Atkin M & Cohen P, "Intelligent Modeling of the User in Interactive Entertainment", pp 8 – 12, AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, Standford, 2002.

[18] Bishop, C.K., Svensen, M., Williams, C.K.I., "GTM: The Generative Topographic Mapping" neural Computation 10(1) pp215-234, 1998.

[19] Buckland M, "Advanced AI Techniques for Game Programming", Chapter 10, Real Time Evolution, Premier Press, 2002.

[20] Cander, L. R., "Ionospheric forecasting technique by artificial neural networks" Electronic Letters 34(16) pp1573-1574, 1998.

[21] Champandard A, "The Dark Art of Neural Networks", AI Game Programming Wisdom, pp 640-651, Charles River Media, 2002.

[22] Charles D, "Enhancing Gameplay: Challenges for Artificial Intelligence in Digital Games", Proceedings of the 1st World Conference on Digital Games 2003, University of Utrecht, The Netherlands, 4-6th November 2003.

[23] Cohen D, "Game Over", New Scientist, pp 47-49, 29th June 2002.