

AI based Game Bot

Aditya Khandare¹, Siddhant Gupta², Rohan Soni³, Manish Joshi⁴, Shabnam Sayyad⁵

^{1, 2, 3, 4, 5}Department of Computer Engineering, Savitribai Phule Pune University

Abstract: Today, we strive to create better and better human beings every day, in every field. Games have existed in human history ever since a long time and are responsible for developing various qualities in humans like strategy developing, decreasing response time etc. So, in this paper, we put forward a method in which a computer will learn how to play computer games using machine learning and AI algorithms and the computer will master the game. We can then use this computer to train humans while allowing them to compete them against the computer and the best part is the computer will constantly be learning the player's moves and getting better and better so as to create a healthy, tough competition for the human.

Keywords: Machine Learning, Artificial Intelligence, Genetic Algorithms, Artificial Neural Networks, Convolutional Neural Networks

I. INTRODUCTION

Any game can be represented as an optimization problem that can be maximized to produce an efficient problem. [2] [10] Due to the advancement in technology and evolution of genetic algorithms like Convolutional Neural Network (CNN), Artificial Neural Networks (ANN), computers can now perform a lot of computationally difficult tasks very easily and optimally. So, we decided to use this computational power to create an expert system that will learn to play computer games. In brief, we will be using CNN to analyse the screen on which the game will run, pixel by pixel. This CNN will then report the current status of the game to the computer [3]. User(human) will have already given the set of inputs the game requires (example: Keystrokes that move the game character or objects). The ANN will then use this set of inputs to play the game. So, using the combination of genetic algorithms [4]; CNN and ANN we can train our bot to play the game and demonstrate how to excel the game or to show how the particular game can be played ideally.

II. RELATED WORK

Since Machine Learning is an emerging field and lot of research is currently being done by various experts of computer field. All the previous works we found we done on a very basic game with limited number of moves and a short play area. Author Jason Lewis has used ANN for solving the game of Tetris. Training a bot for game like Tetris can be done very quickly with today's advancement in technology. [2]. Authors Jose M.Font & Tobias Mahlmann describes in detail how MOBA (multiplayer online battle AREA) supports AI TECHNIQUES in the video game INDUSTRIES. [5] Authors Michael Dann, Fabio Zambetta, John Thangarajah (2018) have used similar technology to develop and train their bot for playing the game INFINITE MARIO. [1]. In this paper, the authors have use Machine Learning algorithms to solve the navigation tasks in Infinite Mario. Again, the objective of this game is simple. The player moves either right or left and completes the maze-like path to reach the goal state i.e. win the game.

III.SCOPE OF THE PROJECT

We will be using the machine learning algorithms and techniques to train a bot which has to perform complex tasks. So basically, the range of inputs will be higher, the objective of the game will be difficult to achieve (as per human standards) and the path that the character has to traverse will be very complex which will again be solved using algorithms like shortest path algorithms.[8]By the end of the project, the developed game bot will not only be able to play the complex game but will also demonstrate how to efficiently complete the game in optimal time span. The bot will be developed such that it has minimal time and space complexities, will be portable i.e. the bot can be transferred from one machine to another without affecting the performance of the bot and will be machine independent. To make the bot efficient, it will be rigorously trained for ample amount of time so that it can overcome any hurdle or new additions in the original game.

IV.GENETIC ALGORITHMS

Like other optimization methods, a genetic algorithm attempt to find inputs from an input space that maximizes the output of some function. In a genetic algorithm, this function is referred to as the fitness function which is used to evaluate the fitness of a candidate input. The algorithm maintains a population of N candidates, where N typically ranges from hundreds to thousands. At the beginning of each iteration of the algorithm, the current population represents a generation of candidates which is used to generate a new population of N candidates for the next generation. Each generation is numbered according to how many iterations of the algorithm have been run Generation 0 is initialized with N random inputs from the input space. A new generation is created by

probabilistically selecting candidates from the current generation. The probability of a candidate being selected is proportional to its fitness evaluated by the fitness function. After all candidates are evaluated, the candidate selection takes place in three phases, which each contributes a certain percentage of candidates to the next generation:

- 1) *Selection*: Candidates are selected and added to the next generation unchanged.
- 2) *Mutation*: Candidates are selected and each candidate's input is slightly altered in some way before being added to the next generation
- 3) *Crossover*: Candidate pairs are selected and then each pair of corresponding inputs are combined in some way to form a new candidate to be added to the next generation. [2]

V. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANN) are systems inspired by the biological neural networks that constitute the animal brains. Neural network is not an algorithm but a framework for many different machine learning algorithms to work together and process complex data inputs. [11] An ANN is based on collection of connected units or nodes called artificial neurons which loosely model the neurons in biological brain. Every connection can transmit a signal from one artificial neuron to the other just like the biological brain.

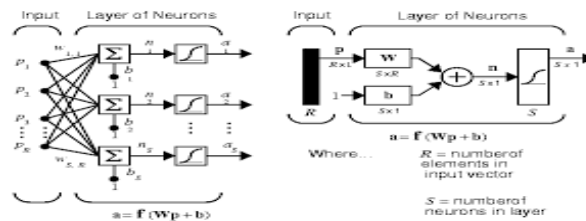


Fig.1: Sample figure where user gives inputs and an output is generated

As in the above figure, ANN consists an input layer where user gives his inputs, an output layer which generates the output after processing the inputs and a number of hidden layers between input and output layers in which all the processing is done.

VI. CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks, in layman's terms can be defined as class of deep learning which can be applied to analyze visual images. CNNs can detect, analyze and generate accurate or near accurate inferences of the images, texts, objects displayed on the screen. The organization of animal visual cortex inspired the development of CNNs in which the connectivity pattern between the neurons resemble the organization of animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual fields known as receptive field. The receptive fields of different neurons partially overlap such that they cover entire visual field. CNN is widely used today in various engineering as well as medical fields in which the goal is to analyze and infer some images.

$$O \left(\sum_{l=1}^d n_{l-1} \cdot s_l^2 \cdot n_l \cdot m_l^2 \right)$$

Fig. 2: shows the sample convolutional network.

Where l is the index of the convolutional layer, d is the depth (number of convolutional layer), n_l is the number of the filters (known as "width") in the l -the layer. n_{l-1} is also known as the number of input channels of the l -the layer. s_l is the spatial size (length) of the filter. m_l is the spatial size of the output feature map. The above Big O notation is the time complexity of all convolutional layers. Using CNN, we would be able to train our bot in exhibiting human like behaviour while playing any generalized game. [6][9]

VII. DEEP Q LEARNING

DQN overcomes unstable learning by mainly 4 techniques.

- 1) Experience Replay
- 2) Target Network
- 3) Clipping Rewards
- 4) Skipping Frames

A. Experience Replay

DNN is easily overfitting current episodes. Once DNN is over fitted, it's hard to produce various experiences. To solve this problem, Experience Replay stores experiences including state transitions, rewards and actions, which are necessary data to perform Q learning, and makes mini batches to update neural networks. This technique expects the following merits: reduces correlation between experiences in updating DNN. Increases learning speed with mini batches. Reuses past transitions to avoid catastrophic forgetting

B. Target Network

In TD error calculation, target function is changed frequently with DNN. Unstable target function makes training difficult. So, Target Network technique fixes parameters of target function and replaces them with the latest network every thousand steps.

C. Clipping Rewards

Each game has different score scales. For example, in Pong, players can get 1 point when winning the play. Otherwise, players get -1 point. However, in SpaceInvaders, players get 10~30 points when defeating invaders. This difference would make training unstable. Thus, Clipping Rewards technique clips scores, which all positive rewards are set +1 and all negative rewards are set -1.

D. Skipping Frames

ALE is capable of rendering 60 images per second. But actually, people don't take actions so much in a second. AI doesn't need to calculate Q values every frame. So, Skipping Frames technique is that DQN calculates Q values every 4 frames and use past 4 frames as inputs. This reduces computational cost and gathers more experiences.

E. Double DQN a Variant of DQN

The main intuition behind Double DQN is that the regular DQN often overestimates the Q-values of the potential actions to take in a given state. While this would be fine if all actions were always overestimating equally, there was reason to believe this wasn't the case. You can easily imagine that if certain suboptimal actions regularly were given higher Q-values than optimal actions, the agent would have a hard time ever learning the ideal policy. In order to correct for this, the authors of DDQN paper propose a simple trick: instead of taking the max over Q-values when computing the Target-Q value for our training step, we use our primary network to choose an action, and our target network to generate the target Q-value for that action. By decoupling the action choice from the target Q-value generation, we can substantially reduce the overestimation, and train faster and more reliably. Below is the new DDQN equation for updating the target value.

$$Q\text{-Target} = r + \gamma Q(s', \text{argmax}(Q(s', a, \Theta), \Theta'))$$

VIII. IMPLEMENTATION

By exploiting the currently available machine learning algorithms to its full extent we will develop a game bot for any 2-D game. It is a 2D graphics game with a number of levels, each level differing from the other in terms of difficulties such as objects that will kill the game character and also differing path complexities. The more you progress in the game, more complex the path becomes, and the number of complexities increase. For navigating our bot through the game environment, path finding algorithms such as A* can be used. [7][8]. In this project, CNN will serve as our eyes and ANN as our brains. CNN will analyze the screen frame by frame, pixel by pixel and report the various objects, position of the character. ANN will perform various computations such as permutations and combinations on the input dataset which will be the keyboard inputs necessary to play the game and find the fittest match/combination of moves which help us reach our goal state [1]. ANN will perform the fitness test and use these fittest of the fittest moves to mutate and generate further moves. The bot will be trained and tested offline on our machines. Training the bot may require anywhere between a few hours to days. More the time the bot is trained more efficiently the bot will work. Our aim is to create a bot that is perfect and can complete the task as quickly as possible and can overcome any change in path or difficulty in the game and will deliver the exact same or nearly same performance even if some changes are made.

IX. CONCLUSION

Through our research we are attempting to combine the CNN and ANN algorithms to help solve optimization problems which can be any generalized goal-oriented game i.e. every game can have different goals or at least a single game where the goal is fixed and remains unaltered throughout the entire game.



REFERENCES

- [1] Michael Dann, Fabio Zambetta , and John Thangarajah, and John Thangarajah, “Integrating Skills and Simulation to Solve Complex Navigation Tasks in Infinite Mario.” , IEEE Transactions on Games (Volume: 10 , Issue: 1 , March 2018)
- [2] Jason lewis , “Playing Tetris with Genetic Algorithms” , Stanford, 2015
- [3] Andreas Precht Poulsen ,Mark Thorhauge ,Mikkel Hvilshj Funch ,Sebastian Risi , “DLNE: A hybridization of deep learning and neuroevolution for visual control” , 2017 IEEE Conference on Computational Intelligence and Games (CIG)
- [4] Chanjuan Liu, Enqiang Zhu, Qiang Zhang, , and Xiaopeng Wei , “Modeling of Agent Cognition in Extensive Games via Artificial Neural Networks.”, IEEE Transactions on Neural Networks and Learning Systems (Volume: 29 , Issue: 10 , Oct. 2018)
- [5] Jose M. Font and Tobias Mahlmann, “The Dota 2 Bot Competition ” , IEEE Transactions on Games (Early Access).
- [6] Michal Kempka,Marek Wydmuch,Grzegorz Runc,Jakub Toczek & Wojciech Jaskowski,”ViZDoom:A Doom-based AI Research Platform for Visual Reinforcement Learning”,Institute of Computing Science,Poznan University of Technology.
- [7] Johan Hagelback,”Hybrid Pathfinding in StarCraft”.Linnaeus University, Sweden.
- [8] Miguel Nicolau, Diego Perez-Liebana, Michael O’Neill and Anthony Brabazon, “Evolutionary Behaviour Tree Approaches for Navigating Platform Games”,IEEE publication 2016.
- [9] Bernd Freisleben,”A Neural Network that learns to play Five-in-a-Row”,IEEE publication.
- [10] Arvind Vijayakumar,”Developing an Artificial Intelligence Bot for Othello”,Bridgewater-Raritan Regional High School.
- [11] Qijin Sun and Suojhu He,”Artificial Neural Network using the training set of dts for pacman game”,IEEE publication 2014.