International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-4, February 2020

Self Learning Gaming Bot using CNN.

Sanket Kine, Mansi Londhe, Akshada Milke, Yadnyesh Modani, Priya Pise.

Abstract: In the present universe of current gaming condition bots are the intelligent agent that assumes a prominent job in the popularity of a game in the market. As these bots have gotten very unsurprising to the games. So here we are proposed an AI model for playing games with high level inputs using reinforcement learning. Algorithm works in the Atari Environment i.e. we are using 2D game. This model consists of the CNN (convolution neural network) for the inputs which is fully connected layers and find out the actions according to the inputs. In this learning -based approach, bots learned how to attack and ignore opponents so that bot can get maximum score. In this learning -based approach, bots learned how to attack and ignore opponents so that bot can get maximum score Then we tried the combine the input method which results maximum score of the bot in the environment for the better performance.

Keywords* : *Machine Learning, Artificial Intelligence, Genetic Algorithms, Artificial Neural Networks, Deep Q Learning , Double DQN Learning.

I. INTRODUCTION

Every game can be represented as an optimization problem that can be maximized to produce an efficient problem. 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. Here we use this computational power to create an expert system that will learn to play computer games. In this system, CNN would be used to analyze 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.

Revised Manuscript Received on February 28, 2020.
***** Correspondence Author

Sanket Kine*, pursing Computer Engineering, Indira College of Engineering And Management, Maharashtra, India.

Mansi Londhe, She is currently pursing Computer Engineering from Indira College of Engineering And Management, Maharashtra, India.

Akshada Milk, pursing Computer Engineering from Indira College of Engineering And Management, Maharashtra, India.

Yadnyesh Modani , Computer Engineering student at Indira College of Engineering And Management, Maharashtra, India.

Dr.Priya Pise, Professor at Indira College of Engineering, Maharashtra, India.

II. RELATED WORKS

Since Machine Learning is an emerging field and lot of research is currently being done by various experts of computer field. Till now all the research done on this was done on very basic game with shorter moves and limited play area. Jason Lewis has used ANN for solving the game of Tetris. Using today's advanced technology, training a bot for game like tertis can be done quickly.

[5] AI techniques in the video game industries are supported by MOBA(multiplayer online battle area) which is described in detail by Jose M. Font & Tobias Mahlmann. [1] Michael Dann, Fabio Zambetta, John Thangarajah (2018) have used similar technology to develop and train their bot for playing the game INFINITE MARIO. 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. RL is machine learning technique where an agent learns to solve a problem while interact with a environment RL algorithm is used to control NPCs in role of playing game.RL algorithm is based on Q learning and it also used for exploration function. It gives rewards based on each action, we get to know the progress of bot in learning RL algorithm is mainly used in robotics domain. 01 to 02 week time window for it.

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. The range of inputs will be higher, the objective of the game will be difficult to achieve (as per human standards) and Shortest path algorithm will be used to traverse the path which will be very complex. By the end of the project, the developed game will show how to efficiently complete the game in limited time span irrespective of the game's complexity. 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. PROPOSED SYSTEM

A. 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 hundreds to thousands candidates which is denoted by N.

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 more fitter the candidate is found by fitness function, chances of getting selected increases proportionally. 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:

- *Selection* – Candidates are selected and added to the next generation unchanged.
- *Mutation* – Candidates are selected and each candidate's input is slightly altered in some way before being added to the next generation
- *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]

Pseudo code GA()

```
initialize population
find fitness of population
   while (termination criteria is reached) do
   parent selection
    perform crossover
   perform  mutation
   survivor selection
   find best
return best
```

B. Artificial Neural Networks

Artificial Neural Networks (ANN) are systems inspired by the biological neural networks that constitute the animal brains. To process complex data inputs and to work together Machine learning algorithms uses Neural Network. An ANN is based on collection of connected units or nodes called artificial neurons which loosely model the neurons in biological brain. Every connection transmits signals from one neuron to another neuron.

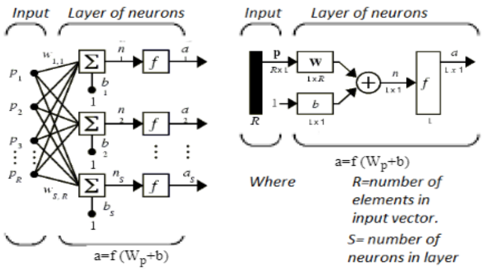

Figure 1

As in the above figure, ANN takes input in input layer of ANN from users then multiple hidden layers between input and output layers perform processing and then the output layer generates the output after processing the inputs.

V. 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. Restricted region is only place where individual cortical neurons respond to stimuli which is of visual field is also refereed as Receptive field. To cover entire field, the receptive fields of different neurons partially overlap on each other. CNN is widely used today in various engineering as well as medical fields in which the goal is to analyze and infer some images.

$$O\left(\sum_{i=1}^{d} n_{i-1}.s_i^2.n_i\,.m_i^2\right) \qquad \text{...............} \ (1)$$

Where i is the index of the convolutional layer, d is the depth (number of convolutional layer). In the i-the layer, ni is the number of the filters (known as "width"). The number of input channel of the i-the layer is known as ni-1. ni is considered as the spatial size(length) of the filter. mi is considered as the feature map's spatial size. 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 behavior while playing any generalized game.[6][9]

VI. DEEP Q LEARNING

DQN overcomes unstable learning by mainly 4 techniques.

- Experience Replay
- Target Network
- Clipping Rewards
- Skipping Frames

- **Experience Replay**

DNN is considered as easily overfitting current episodes. It is hard to produce various experiences after the overfitting of DNN is done. In order to solve this problem, stores experiences including state transitions, rewards and actions are stored by Experience Replay, which are necessary data to perform Q learning and mini batches are made to update neural networks. This technique expects the following merits: Correlation between experiences between updating DNN is reduced. Using mini batches, speed of learning increased. Reuses past transitions to avoid catastrophic forgetting.

Target Network

In TD error calculation, target function is changed frequently by using DNN.

Unstable target plays very important role in making training difficult. So, Target Network technique fixes parameters of target function and replaces them with the latest network every thousand steps.

- **Clipping Rewards**

There is different score scales in each game. For example, in Pong, players can get 1 point by wining the play and -1 point for loosing. However, in Space Invaders, players get 10 to 30 points by defeating invaders. This difference makes training unstable. Thus, Clipping Rewards technique clips scores, all positive rewards are set +1 and rest are set -1.

- **Skipping Frames**

ALE is capable of rendering at high rate of 60 images per second. But practically, people don't take actions so much in a second. Calculating Q values every frame won't be done by AI. 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.

Pseudo code of DQN()

```
Initialize Q_{0}(s,a) for all pairs (s,a)
s = initial state
k = 0
while(convergence is not achieved)
{
   simulate action a and reach state s'
   if(s' is a terminal state)
   {
      target = R(s,a,s')
   }
   else
   {
  target = R(s,a,s') + γ́max_{a'}Q_{k}(s',a')

   }
$\theta_{k+1} = \theta_k - \alpha\Delta\ \Delta_\theta$Es' p(s'|s,a)[($Q_\theta$(s,a)-target(s'))^{2}]|$_{\theta = \theta_k}$
          s = s'
}
```

A. Double DQN a variant of DQN

To avoid the overestimation of Q-values of the potential actions to given state by regular DQN, Double DQN is used. While this would be fine if all actions were always overestimating equally, there was reason to believe this wasn't the case. In case sub-optimal actions were provided with higher Q-values regularly as compared to optimal actions, the agent might face hard time learning the ideal policy. Instead of taking the max over Q-values while computing the target Q-value for our training step, we will use our primary network to choose an action and our target network to generate the target Q-value for that action.

This simple trick is proposed by the authors of DDQN paper. We can substantially reduce the overestimation, and train faster and more reliably by decoupling the action choice from the target Q-value generation. Following is the new DDQN equation for updating the target value.

Q-Target = r + γQ(s',argmax(Q(s',a,ϴ),ϴ')) (2)

VII. IMPLEMENTATION

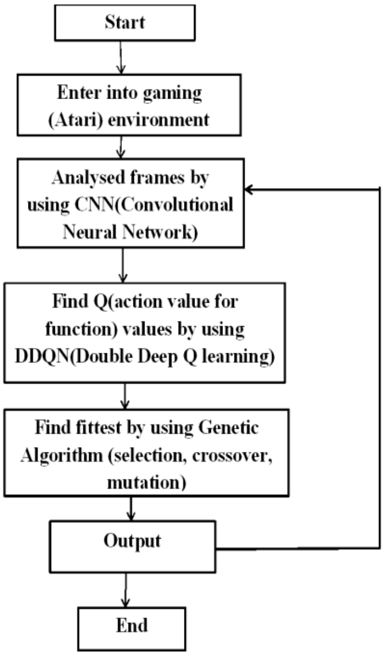
Start
Enter into gaming (Atari) environment
Analysed frames by using CNN(Convolutional Neural Network)
Find Q(action value for function) values by using DDQN(Double Deep Q learning)
Find fittest by using Genetic Algorithm (selection, crossover, mutation)
Output
End

Figure 2. Flowchart

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. The dataset 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 will be nearly perfect and will be able to complete the given task as quickly as possible efficiently. The bot will be able to 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.

VIII. RESULT AND DISCUSSION

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. Table below shows the result of using CNN, ANN and DDQN in gaming bot.

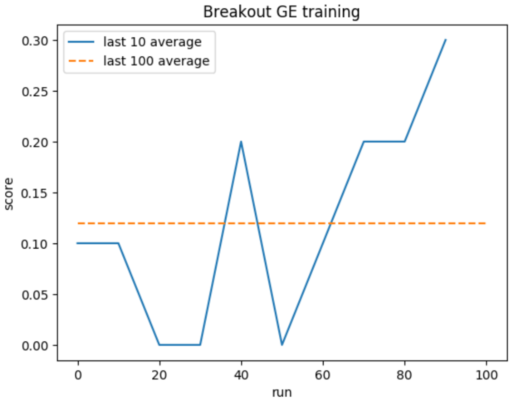

Figure 3 Training phase of bot

Fig.3 shows that comparison of the bot or agent after the 100 iteration in the game of Breakout by using the CNN, genetic algorithm in the Atari environment. Parameter of the comparison are score vs run.

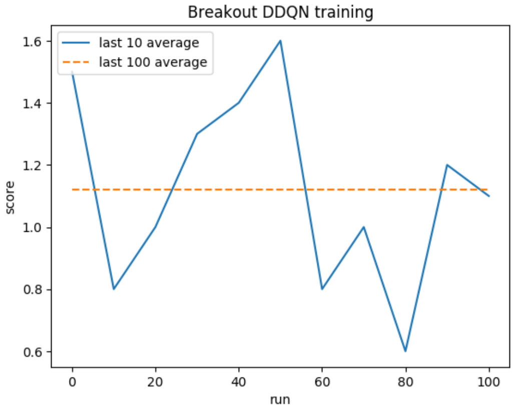

Figure 4. Testing phase of bot

Fig.4 shows that comparison of the bot or agent after the 100 iteration in the game of Breakout by using the CNN, DDQN, genetic algorithm in Atari environment.

- **Algorithms VS Parameters**

Table I : Summary of algorithms which we used in Model

Parameters	Algorithms	
	CNN	RNN
Powerful	More	Less
Size	Fixed input /output	Arbitrary input /output
Idle	Images and video	Text and speech
Data	Spatial	Temporal
Connectivity	Connectivity Patterns between neurons	Recurrent neural networks use time –series

Table II: Summary of algorithms

Algorithms	Parameters	
	Solution	Select move
Hill	Local Optimal solution	randomly
A*	Local Optimal solution	Smallest weight
Genetic	Global Optimal solution	Fittest individuals

IX. CONCLUSION

In this paper, we discussed about the learning algorithms for self-learning gaming bot. Developing AI agent can deal with the complexities of the gaming environment, but it can help in the other areas like entertainment, education and training. Our main goal is to create a self-learning Gamming bot. we started with the CNN and DDQN algorithm but the result of this algorithm is it result as it slowdowns the learning so after the comparison between the A* Algorithm , Hill climbing and Genetic algorithm we get to know that genetic algorithm speed ups the learning. This algorithm is generalised algorithm in which we can apply for 20+ games in the Atari environment.

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. Priya Dudhale- Pise , Dr. Nilesh J. Uke, "Hybrid Deduplication for Secure Data Sharing UsingErasure Techniques and HIDC" year 2017.
5. Priya Dudhale- Pise , "Content-Based Deduplication Of Data Using Erasure Technique For Rto Cloud" year 2018.

AUTHORS PROFILE

Sanket Kine, He is currently pursing Computer Engineering from Indira College of Engineering And Management . He has interests in the fields of Machine Learning and Artificial Intelligence. He has won SIH-2019 and now he is going to works on the machine learning

Mansi Londhe, She is currently pursing Computer Engineering from Indira College of Engineering And Management. She has interests in the fields of Machine Learning and deep learning .

Akshada Milk ,Being a computer engineer her interests is always towards the new technologies and their algorithms. She is currently pursing Computer Engineering from Indira College of Engineering And Management She is interested in the machine learning and the deep learning

Yadnyesh Modani , He is currently a Computer Engineering student at Indira College of Engineering And Management He has interests in the fields of Machine Learning, and data science. He has won SIH-2019 and now he is going to works on the data science

Dr.Priya Pise She received her Ph.D. in Computer Engineering from JJT University. She is currently working as a Professor at Indira College of Engineering. Her research interests are mainly in the areas of cloud computing, big data, networks security, and cyber-crime systems. She has published over 50+ conference and journal papers.