International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 08 | Aug 2021 www.irjet.net p-ISSN: 2395-0072

SUPERVISED LEARNING VS REINFORCEMENT LEARNING ON GAME ENVIRONMENTS

Amogh Sawant[1]

[1]*Student, Information Technology Department, Atharva College of Engineering, Maharashtra, India*

Abstract - *Artificial Intelligence started gaining popularity in the 1900s and since then, there are a lot of different ways to train Artificial Intelligence. Some may be concerned about the accuracy of the given task, others maybe are concerned with the efficiency of the learning process, and so on. In this paper, we hope to perform and analyze two different types of learning techniques in a relatively simple game environment and draw conclusions based on which algorithm provides us with better results. Here, we will be using Flappy Bird as our game environment and compare Supervised Learning and Reinforcement Learning (using Genetic Algorithm).*

Key Words: **Artificial Intelligence, Supervised Learning, Reinforcement Learning, Game Environment, Genetic Algorithm**

1. INTRODUCTION

Science fiction shows and comics familiarized the world with the concept of Artificial Intelligence (AI). While most people think that AI is only used in super-advanced computers and applications, it is very ubiquitous and is used very often even in simple applications like camera viewfinders and entertainment applications like video games. Focusing on the latter part, AI and video games are the two industries that have been going hand in hand for many years now. Video games industries make use of AI to produce responsive and intelligent behavior in the sprites (2D bitmap integrated into a larger scene) which may represent the player itself, as enemies or companion of the player, depending on the game. Game companies used to utilize if-else statements to perform certain actions, however, these days they have been shifting towards the use of AI to generate more natural and logical reactions in the sprites.

There are different types of learning algorithms, for instance, Supervised Learning, Unsupervised Learning, Reinforcement Learning, etc., but according to most studies, gaming companies often use supervised learning and reinforcement learning in video games. Hence, we will perform a research experiment comparing these two learning algorithms. Each of these algorithms has its own set of advantages and disadvantages which we will try to demonstrate using the flappy bird game environment. Flappy Bird is a game where the bird, controlled by the player, has to navigate through an infinite array of pipes with some space between them. The location of these empty spaces, which we will refer to as gaps in this paper, are randomly located throughout the series of these pipes. The game accepts only one input, the command of "jump", which makes the bird jump in the air while the force of gravity tries pulling the bird down again. The objective of the player is to try passing through as many pipes as possible without colliding with the ceiling (top edge of the screen), ground (bottom edge of the screen), and the pipes.

Figure 1 Original Flappy Bird

2. LITERATURE REVIEW

Sebastian Risi , Julian Togelius [1] – This excellent paper explains the idea of neuroevolutionary algorithms which can be used on simple 2D video games. There are different Neuroevolutionary algorithms like Genetic Algorithm, NeuroEvolution of Augmented Topologies, popularly known as NEAT, etc. They compare the performance of different algorithms and analyze their outcomes. They also highlight the challenges that come with designing a fitness function for these algorithms as it plays a crucial role in these games.

3. HARDWARE & SOFTWARE USED

3.1 Hardware

- CPU: Intel Core i5-8400
- GPU: Nvidia GTX 1050
- Memory: 8 GB RAM

3.2 Software

- IDE Used: VS code
- OS version: Windows 10 64-bit
- Modules: TensorFlow, Keras
- Languages Used: Python
- Environment: Anaconda

4. PROPOSED SYSTEM

For the experiment, we created a more toned-down and simpler version of the original game. The agent, which will be controlled by the Neural Networks (NN), represents the bird and the black thin rectangles represent the pipes from the original game. Note that whenever we mention 'flappy bird' in this paper from now on, we talk about our version (Fig. 2) of the original game.

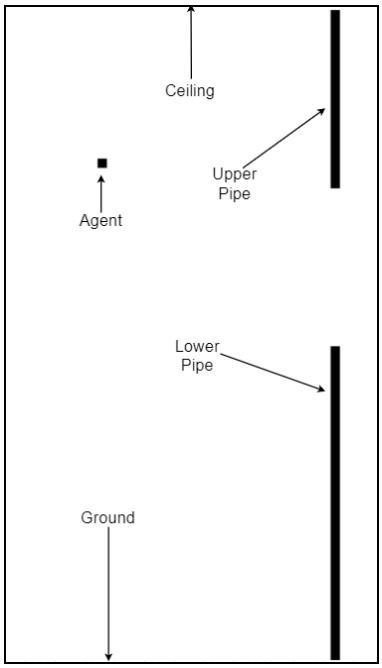


Figure 2 Our Version of Flappy Bird

4.1 System Architecture

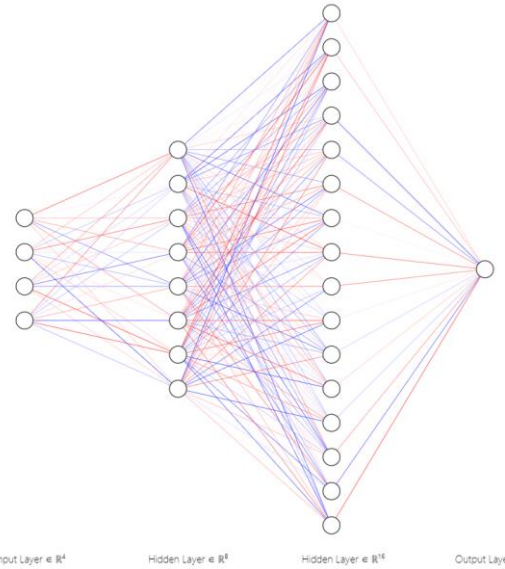


Figure 3 NN Architecture we used for Genetic Algorithm

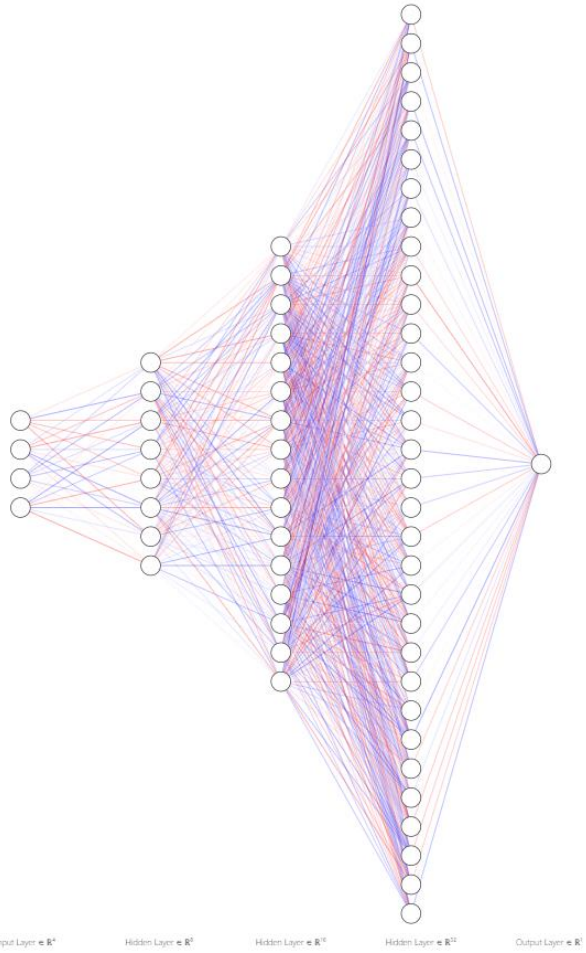


Figure 4 NN Architecture we used for Supervised Learning

Genetic Algorithm

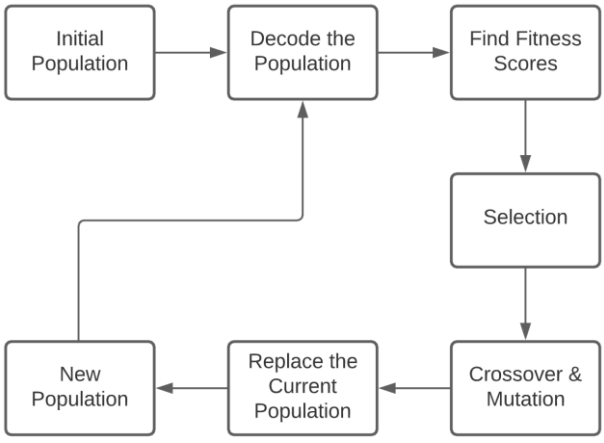


Figure 5 Genetic Algorithm Flow Chart

Genetic Algorithm is a method of training and/or optimizing the given task by using the concept of natural evolution. In this, we create a random generation at the start which we hope to optimize to produce the best outcome for the given task and it does this by exploring the environment. As stated

before, we are using Neural Network as the encoder and the weights and biases of the neural network are the hyperparameters that Genetic algorithm tries to optimize based on a fitness function. A fitness function is used to find the best genes (weights and biases) by calculating the fitness score of each agent. The best NN of each generation, having the highest fitness score, is then crossover to produce the next generation. We hope of creating a better generation than their predecessor because all the agents in the current generation will be having the qualities of the best NNs from the last generation. We also mutate the genes to keep the randomness in the process and attempt to find even better agents.

Supervised Learning

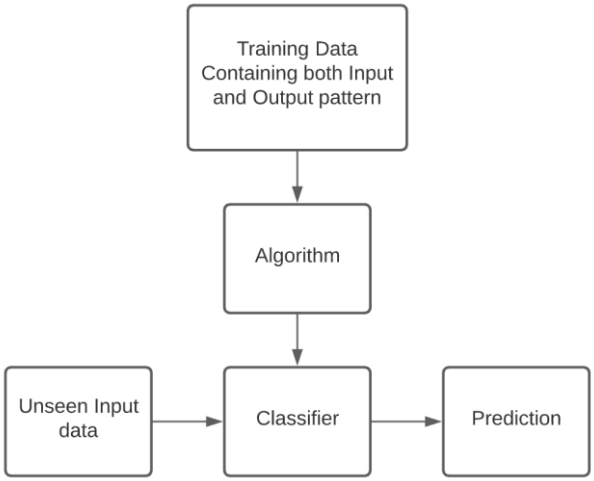


Figure 6 Supervised Learning Flow Chart

In supervised learning, the machine is trained using a well labelled data and the machine predicts the output based on that training data. The labelled data here refers to the input data which is already marked with the correct output. The provided training data acts as a supervisor and teaches the machine to predict the correct output. The aim of the supervised learning algorithm is to find a mapping function to map the input variable with the output variable.

The model (the machine) is trained until it successfully identifies the association between the input and the output data. Whenever the model makes a wrong choice during the training, an error is calculated based on the choice it made and how far it was from the correct output. This error is then used to make some specific changes and readjust the model to correctly identify the choice. This process is repeated with all the training data until the model is successfully able to predict the outcome with the unseen, but similar data.

5. METHODOLOGY

We use Neural Network as our encoder as mentioned while letting the two competing learning techniques to optimize the weights and biases of the neural network. The input layer takes 4 inputs as follows:
1. X coordinate of the agent.
2. Y coordinate of the agent.
3. Distance between the agent and the lowest part of the upper pipe.
4. Distance between the agent and the highest part of the lower pipe.
The output layer has only one node which makes the decision of jumping or not jumping.

5.1 Training Process of Genetic Algorithm

We programmed the reward system so that whenever an agent successfully passes between the pipes, it gets a reward of +200, while the fitness function was 1 / (distance between agent and center of the gap between pipes) which was then added cumulatively. We choose this so that whenever the agent approaches near the center of the gap (because that's the ideal way of passing through the gaps), it gets a greater reward and vice-versa.

Steps for selection of best agent using Genetic Algorithm:

1. Initialize a population of around 100 agents, having same NN architecture, but random weights and biases.

2. Feeding the input data to the neural networks.

3. Use fitness function as a way to reward the agent. In this case Fitness function = 1 / (distance between agent and center of gap between pipes).

4. Select a few agents having highest fitness score from the population and perform crossover and mutation on them to generate the next generation.

5. We continued this process for over 30 generations.

6. Test the NN model to see how good it performs.

7. Save the model for the future use.

5.2 Training Process of Supervised Learning

We create the training data by playing the game for quite a while and then normalizing the data by capturing around 50,000 frames where we made the agent jump and 50,000 frames where we didn't. In total, we capture the data of around 100,000 frames, using 75,000 frames as training data, and the remaining frames as testing data.

Steps for Training of Supervised Learning:

1. Import the NN model APIs such as Keras and TensorFlow.

2. Build a feedback model with enough nodes and layer. Our NN architecture is shown in the Fig. 4.

3. Feeding the input data in the input layer.

4. Feed the training data and the testing data that we created before.

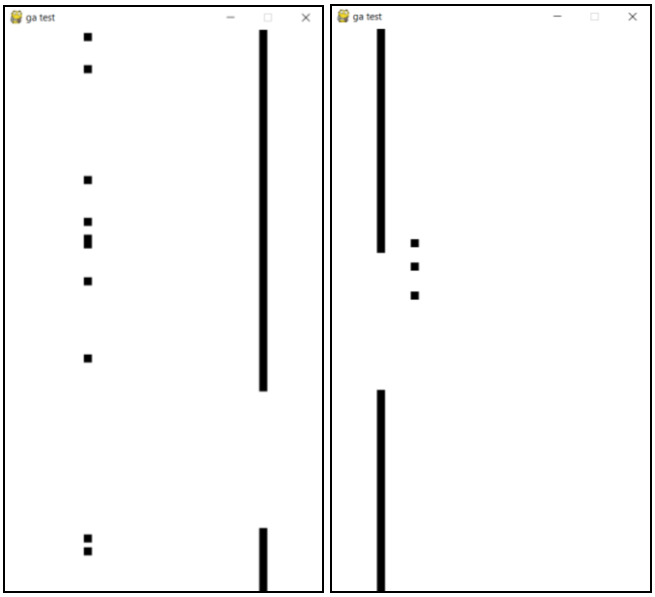
5. Train the model for appropriate number of epochs.

6. Testing the model.

7. Save the model for future uses.

6. RESULTS

6.1 Result Analysis of Reinforcement Learning (using genetic algorithm) on Flappy Bird



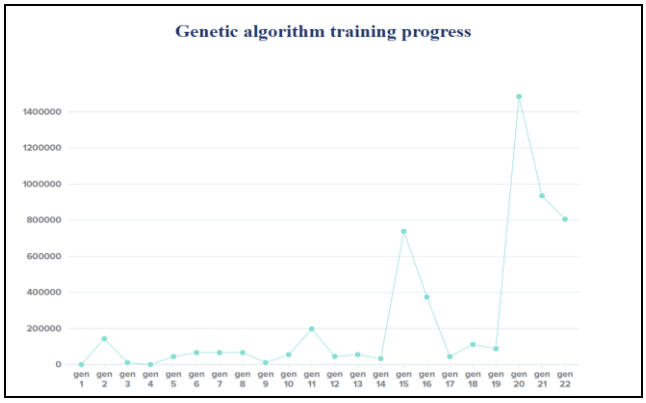


Figure 7 Flappy Bird Training on Genetic Algorithm

The training process took around 15-20 minutes before the agent (bird) mastered the game and started navigating through the gaps indefinitely. The agent (bird) was trained on the gap size of 200 units. While testing, we changed the gap size from 120 units to (>250) units. We found that the agent is easily able to pass through the gap size greater than 200 units. In contrast, when we started decreasing the gap between the pipes, we observed that the agent was able to perform fairly well through the range of 150 units to 190 units of gap size, but it started to struggle when the gap size was (<150) units as it became too narrow. In addition to that, we increased the gravity of the environment by twice than it was trained on, forcing the bird to make more jumps to pass through the gaps, and it surprisingly performed well. Again, it fails to pass successfully through the gaps when we make it too extreme for the agent by increasing the gravity any further. Note that while changing the gravity, the gap size was constant at 200 units.

6.2 Result Analysis of Supervised Learning on Flappy Bird

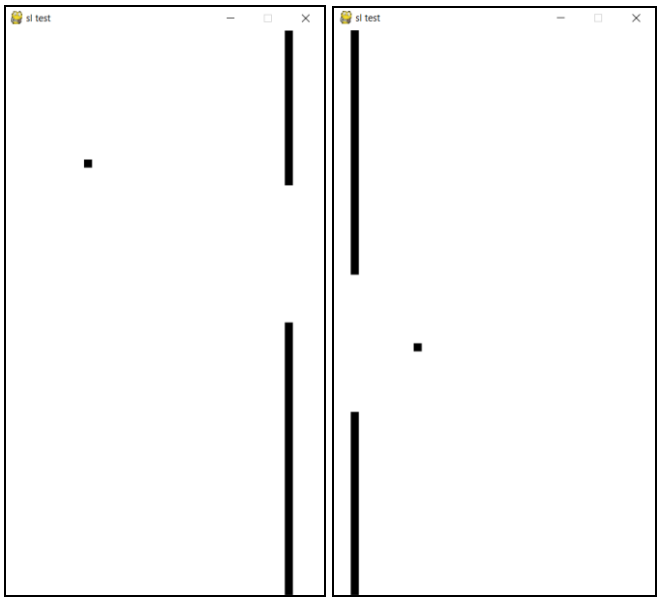


Figure 8 Flappy Bird training on Supervised Learning

The training process took almost 2 mins to achieve around 92% accuracy. The agent (bird) was trained very well on 100,000 frames of data with a gap size of 200 units. After training, the model was successfully able to navigate through the gaps between the pipes indefinitely.

We perform the same changes to the environment again as mentioned before, i.e., changing the gap size between the pipes and the gravity of the environment. Here, we observe that the agent was able to pass through the pipes of a gap size of 180 units to 230 units. After reducing the gap size to (<180) units the agent starts to colliding with pipes, whereas increasing the gap size to (>230) units, confuses the agent and it doesn't take any actions, resulting in the agent falling on the ground due to the gravity. On the other hand, increasing gravity to more than 1.5 times than the gravity it was trained on, the agent fails to pass through the gaps.

7. CONCLUSION

To conclude, we found out that genetic algorithm gives us better performance and is more adaptable to changes in the environment compared to supervised learning. However, the training time of Reinforcement Learning is significantly more than Supervised Learning. Hence, if the game has more unpredictable scenarios, where the sprite has to adapt to the environmental changes very quickly, we would advise using Reinforcement Learning with Genetic Algorithm. In contrast, if the game has more predictable scenes, Supervised learning would be a more efficient choice.

It is also important to note that since supervised learning algorithms use existing data to train on, which in this case humans are the ones generating it, it will always perform slightly worse than the creator of the data because it is impossible to achieve 100% accuracy. On the other hand, reinforcement learning using genetic algorithm explores the environment by itself and thus, can be a lot better than humans.

REFERENCES

[1] S. Risi and J. Togelius, "Neuroevolution in Games: State of the Art and Open Challenges," in IEEE Transactions on Computational Intelligence and AI in Games, vol. 9, no. 1, pp. 25-41, March 2017, doi: 10.1109/TCIAIG.2015.2494596.