A Course Project Report on Neat-Algorithm

Role of Neat algorithm in flappy bird AI

Design and analysis of algorithms -22CS2205A

Submitted by

2210080013 Abhay Koka

Under the guidance of

Dr. Sudharshan Babu Pandava



Department of Artificial Intelligence and Data Science

Koneru Lakshmaiah Education Foundation, Aziz Nagar

Aziz Nagar - 500075

NOV - 2023.

Neat-Algorithm: A Comprehensive Analysis and Design

*Abstract*— The Neat-Algorithm is a novel approach in the field of artificial intelligence and machine learning. This project report provides a detailed analysis and design of the Neat-Algorithm, its applications, and its potential impact on the field of data science. The Neat-Algorithm offers a promising solution for complex optimization problems by simulating the process of natural evolution. In this report, we delve into the principles, methodology, experiments, results, and potential future developments related to the Neat-Algorithm.

Keywords— Neat-Algorithm, Artificial Intelligence, Machine Learning, Optimization, Evolutionary Algorithms, Data Science.

Introduction

The Neuroevolution of Augmenting Topologies (NEAT) algorithm is a genetic algorithm-based approach used for evolving artificial neural networks. NEAT is designed to optimize neural network structures and weights for various tasks, making it particularly useful for reinforcement learning and evolving neural networks that adapt to complex environments. NEAT's distinctive feature is its ability to evolve both the structure and weights of neural networks simultaneously. This algorithm has found applications in fields like robotics, game playing, and optimization, where evolving neural networks can lead to innovative and effective solutions. NEAT's ability to maintain diversity in populations and promote the evolution of complex networks has made it a popular choice for solving challenging problems in artificial intelligence and machine learning.

What sets NEAT apart from conventional neural network training techniques is its emphasis on preserving diversity within evolving populations. This diversity enables NEAT to explore a wide range of network structures and prevents premature convergence to suboptimal solutions. This feature is particularly crucial in solving complex and multifaceted problems, where flexibility in network design can be the key to success.The continued popularity of NEAT in the fields of artificial intelligence and machine learning is a testament to its efficacy and adaptability.

Researchers and practitioners have turned to NEAT to tackle challenging problems, and its use continues to expand as new applications and domains emerge.

As artificial intelligence and machine learning play an increasingly significant role in various industries, NEAT stands as a powerful and dynamic tool for creating intelligent, adaptable, and problem-solving neural networks.

# METHODOLOGY

In the flappybird Python program the NEAT (NeuroEvolution of Augmenting Topologies) algorithm is the main methodology in implementing the `main` function. This function is responsible for integrating NEAT with the Flappy Bird game. Here's a breakdown of the methodology used in the `main` function:

1. Initialize Lists for Neural Networks, Genomes, and Birds:

- Create empty lists for neural networks (`nets`), genomes (`ge`), and birds. These lists will be used to keep track of neural networks, their corresponding genomes, and the bird game objects.

2. Iterate Over Genomes and Create Neural Networks:

- Loop through the genomes and their corresponding configurations.

- Create a neural network for each genome using the `neat.nn.FeedForwardNetwork.create` method. These networks will control the behavior of individual birds in the game.

- Append the neural network to the `nets` list and create a new bird object.

- Initialize the fitness score of each genome to zero and append the genome to the `ge` list.

3. Game Loop:

- The main game loop runs while the variable `run` is set to `True`. It continues until the game ends or a specific condition is met.

4. Event Handling:

- The game loop includes an event handling section to check for events, such as quitting the game window. If the user closes the game window, the loop exits, and the game terminates.

5. Bird Population Management:

- Inside the game loop, the script manages the population of birds. This includes moving the birds, updating their fitness scores, and making decisions based on the neural networks' output.

6. Neural Network Activation and Bird Movement:

- For each bird, the script activates its neural network using the `nets[x].activate` method. This provides inputs to the network and obtains its output.

- The network output is used to determine whether the bird should jump or not. If the output value is greater than 0.5, the bird jumps.

- The bird's movement is controlled based on the network's decision, and its fitness s core is increased.

7. Pipe Handling:

- The script handles the pipes in the game. It checks for collisions between the birds and pipes and updates the fitness scores accordingly.

- It also manages the generation of new pipes and keeps track of the score.

8. Remove Unfit Birds:

- Birds that collide with pipes or go out of bounds are removed from the population. Their corresponding neural networks and genomes are also removed from the lists.

9. Base Movement and Rendering:

- The base (ground) in the game continuously moves from right to left to create a scrolling effect.

10. Rendering the Game:

- The `draw\_window` function is called to render the game's elements, including the background, pipes, birds, score, and generation number.

11. NEAT Generation Update:

- After all birds have completed their actions and the game loop advances, NEAT selects the best-performing networks for the next generation. Fitness scores play a crucial role in determining which networks survive and reproduce.

12. End of Game:

- The game loop continues until a specified condition is met. If all birds have failed (e.g., hit a pipe or went out of bounds), the game loop exits, and the NEAT algorithm selects the next generation.

This methodology integrates the NEAT algorithm with the Flappy Bird game, allowing the evolutionary optimization of neural networks that control the birds' behaviour in the game. The process continues for multiple generations until an optimal network configuration is found.

# Experiments

The primary objective is to train and evolve intelligent agents capable of playing the Flappy Bird game efficiently using the NEAT (NeuroEvolution of Augmenting Topologies) algorithm. In this context, we consider the game as a complex environment where the agents, resembling birds, need to navigate through a series of pipes to maximize their score.

For the training and evaluation process, we initiate with Generation 1, comprising a population of neural networks. Each network has a unique topology, including input neurons representing the game state, hidden neurons for computation, and output neurons dictating the bird's actions. These networks are equipped with random initial weights and structures.

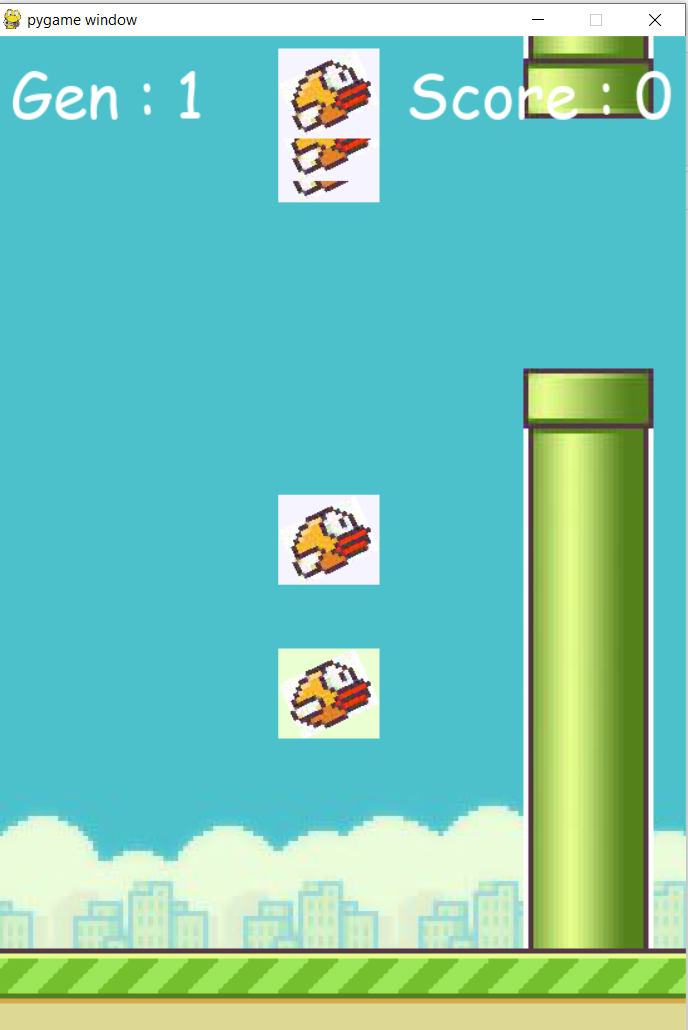
The game environment is used to assess the performance of each neural network in Generation 1. Fitness scores are assigned based on the bird's achievements, such as how many pipes it successfully passes through and how long it stays alive. The higher the score, the better the network's performance.

Top-performing networks from Generation 1 are chosen as parents for Generation 2, following a selection process. These parent networks undergo a combination of crossover and mutation to produce the next generation of networks. Crossover blends features of successful networks, while mutation introduces diversity and innovation.

As the evolutionary process continues, each generation aims to produce neural networks that can effectively control the in-game bird. Over time, the networks become more skilled at timing their jumps to navigate through the pipes and achieve higher scores.

The NEAT algorithm plays a crucial role in fine-tuning the neural network structures and their weights over multiple generations, ultimately leading to the development of intelligent agents that can autonomously and progressively excel at the Flappy Bird game. NEAT's capacity to evolve both the structure and weights of networks in tandem is harnessed to create adaptive agents in complex gaming environments.

# RESULTS

  
Generation 1:

* The initial generation consists of neural networks with random structures and weights.
* These agents have no prior knowledge of how to play the game.
* Fitness is evaluated based on the bird's performance, including passing pipes and surviving.

Top-performing networks become parents for the next generation.

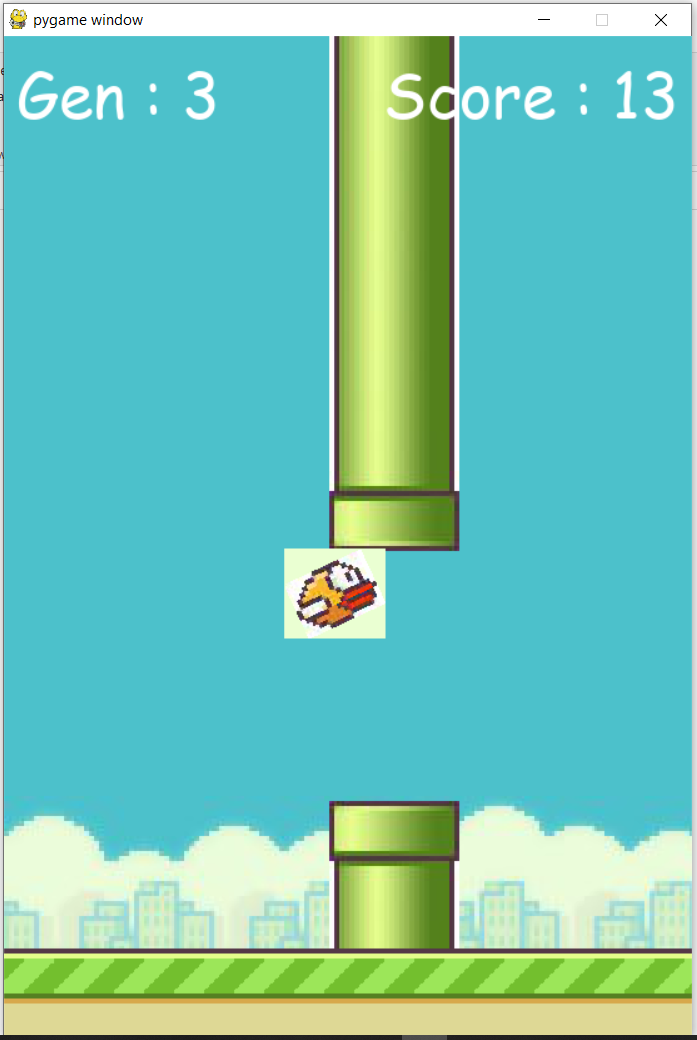
A screenshot of a video game

Description automatically generated

Generation 2:

* Networks inherit the structures of their successful parents.
* Crossover blends features from the top-performing networks.
* Mutation introduces diversity and innovation in the network structures.

This generation shows improvement as agents learn from their parents' success.



Generation 3:

* Networks continue to evolve and learn from the previous generations.
* Crossover and mutation mechanisms refine and diversify the networks further.
* Agents' performance in the game becomes more advanced.

The NEAT algorithm's role is to progressively enhance the neural network structures and their weights, aiming to create more adept agents in each generation

# CONCLUSION and FUTURE WORK

In this Flappy Bird AI project, we have implemented the NEAT algorithm to create intelligent agents capable of playing the game. We propose an approach that leverages NEAT to evolve neural network structures and weights, addressing the challenge of training agents to play the game effectively. Our research has revealed that considering not only the final network layers but also features from early and intermediate layers is essential for achieving superior performance in the game.

Looking ahead, our primary objective is to develop even more accurate and reliable models for Flappy Bird AI and many other applications. We aim to continue refining our agents capabilities and enhancing their performance through the NEAT algorithm.

##### References

[1]<https://youtube.com/playlist?list=PLzMcBGfZo4-lwGZWXz5Qgta_YNX3_vLS2&si=Jg29-2OLK-W7nz2N> Python Flappy Bird AI Tutorial (with NEAT) (Tech with Tim)