Snake Game AI Using Deep Neural Networks

MEMBER 1
KETHACHAITANYA
REGNO:22BCE9090
SLOT:A2

MEMBER 2(TEAM LEADER)
S.LALITH KUMAR
REG NO: 22BCE9938
SLOT: A2

MEMBER 3:
P.VARUN
REGNO:22BCE7303
SLOT:A2

***Abstract*—**

The evolution of artificial intelligence (AI) in game-playing has led to significant advancements in decision-making and autonomous control. This project explores a Deep Neural Network (DNN) approach to playing the classic Snake game using supervised learning, as an alternative to reinforcement learning techniques like Deep Q-Networks (DQN). The proposed method follows five stages: (1) implementing the Snake game using pygame, (2) collecting gameplay data from a rule-based agent, (3) training a DNN model to predict optimal moves based on game states, (4) deploying the trained model to play the game, and (5) evaluating its performance.The training dataset is generated using a heuristic-based bot, and a fully connected feedforward neural network is trained using cross-entropy loss and the Adam optimizer in PyTorch. The trained model generalizes well to unseen game scenarios, demonstrating intelligent movement decisions that improve gameplay performance. Results indicate that a supervised learning-based DNN can effectively replace reinforcement learning methods, offering a computationally efficient and stable solution.This work highlights the feasibility of DNNs for autonomous gameplay and opens avenues for further enhancements, including convolutional networks for visual inputs and hybrid approaches that integrate supervised and reinforcement learning for adaptive decision-making.

I. Introduction (*Heading 1*)

Artificial Intelligence (AI) has improved tremendously over the past few years, especially in the field of game-playing AI, with deep learning methods ushering in revolutionary breakthroughs. AI systems have been able to beat human players in traditional board games like Chess, Go, and Shogi, as well as intricate computer games like Dota 2 and StarCraft II. Typically, game-playing AI uses Reinforcement Learning (RL) methods, whereby an agent learns by interacting with the environment and adapting a reward-based policy. But reinforcement learning techniques, like Deep Q-Networks (DQN), require large amounts of exploration, computation, and training time and are therefore unsuitable for easier tasks.The Snake game, an arcade game, is an intriguing problem for AI because it has a dynamic environment and decision-making constraints. The game asks for an agent to move in a Bounded grid, gather food, and evade crashes while accumulating its score. Most current AI solutions for the Snake game utilize Q-learning or DQN-based reinforcement learning to deliver optimal performance. Although these work well, they tend to use millions of iterations, hyperparameter tuning, and high computational effort, which might be wasteful for a comparably straightforward game environment.

To overcome these issues, this study investigates a different solution: employing a Deep Neural Network (DNN) learned through supervised learning to play the Snake game independently. Rather than learning through trial and error as in reinforcement learning, the new method is to train a DNN on pre-obtained gameplay data created by a rule-based bot. The main benefits of this method are:

Computational Efficiency: Unlike reinforcement learning, which requires continuous exploration and policy updates, supervised learning allows the model to directly learn from labeled data, reducing training time.

Stability and Generalization: The learned model picks up regular decision-making habits from organized data, resulting in more stable and predictable performance.

Simplified Training Process: DNN training with labeled examples avoids the requirement of intricate reward engineering or hyperparameter tuning, which are crucial in reinforcement learning.

II. Literacture survey

A. Overview of the Snake Game in AI Research

The Snake game is a widely used benchmark in AI research due to its **simple rules yet challenging dynamics**. The game's objective is for the snake to navigate the grid, collect food, and avoid collisions, making it a dynamic sequential decision-making task.

Various AI techniques have been applied to the Snake game, including:

B. Rule-Based Approaches

Rule-based techniques were the earliest AI methods applied to the Snake game. They employ pre-established logic to decide based on the state of the game. Some common techniques are:

- Greedy Algorithms: Head towards the food in the direction of the shortest Euclidean distance.
- Hamiltonian Cycles: Specialized paths that ensure the snake never gets trapped (Salvador et al., 2009) [1].
- A* Pathfinding Algorithm: Determines the shortest path to the food source without hitting obstacles.

Although rule-based approaches guarantee determinable behavior, they do not support adaptability and generalization in complicated situations.

C. Reinforcement Learning Approaches

Reinforcement Learning (RL) has been extensively used in the Snake game, enabling the agent to acquire policies by trial-and-error interaction with the environment. The most popular RL methods are:

- Q-Learning: A tabular approach where the agent learns state-action values (Q-values) by exploration (Watkins, 1989) [2].
- Deep Q-Networks (DQN): Proposed by Mnih et al. (2015) [3], DQN merges Q-learning with deep neural networks to estimate Q-values in high-dimensional state spaces.
- Double DQN (DDQN): Van Hasselt et al. (2016) [4] introduced DDQN to overcome the overestimation bias of DQN.

While successful, RL-based approaches have a number of drawbacks:

- High computational cost
- Long training durations
- Instability of training
- Hyperparameter sensitivity

D. Supervised Learning in Game AI

Supervised Learning (SL) has been investigated as a substitute for reinforcement learning for game AI in recent research. Supervised learning differs from RL in that it trains the model with pre-existing labeled data, and hence it is more efficient and stable. Supervised learning has been utilized in numerous game-playing tasks, such as:

- Chess Move Prediction (Silver et al., 2018) [5]: Supervised learning was employed to pre-train AlphaZero's policy network prior to reinforcement learning.
- Atari Games (Hussein et al., 2017) [6]: Showed that imitation learning using supervised data was able to obtain competitive performance on a number of Atari games.
- Pac-Man (Sharma et al., 2020) [7]: An expert-trained DNN played Pac-Man successfully without reinforcement learning.

These works imply that supervised learning is most appropriate for well-structured, rule-based games such as Snake where labeled data is easily obtainable.

E. Deep Neural Networks for Game AI

Deep Neural Networks (DNNs) have been extensively utilized in game-playing AI because they can approximate complicated functions. DNNs have been used in both supervised and reinforcement learning. The architecture is usually composed of fully connected layers or convolutional layers based on the input representation.

In the case of the Snake game, feedforward networks with full connectivity are usually employed to translate game states into best actions. The most significant strengths of DNNs for game AI are:

- High expressiveness
- Generalization ability
- Effective training on labeled data

Nevertheless, the majority of current DNN-based methods are based on reinforcement learning, with few attempts to explore supervised learning techniques. This work seeks to bridge this gap by proving the efficacy of supervised learning-based DNNs for Snake game AI.

F. Comparative Analysis: Supervised Learning vs. Reinforcement Learning

Feature	***Supervised Learning (DNN)***	***Reinforcement Learning (DQN)***
Training Time	Fast	Slow (Millions of iterations)
Computational Cost	Low	High
Stability	Stable	Unstable
Generalization	Limited (depends on dataset)	High
Data Requirement	Pre-collected labeled data	Self-generated

This comparison highlights that **supervised learning** offers **faster training and greater stability**, making it a suitable choice for games like Snake with deterministic rules.

III. Methodology

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

A. *Game Environment*

The Snake game environment is implemented using the Pygame library, which provides a structured framework for game development. The game consists of a grid where the snake moves to collect food while avoiding collisions with itself and the boundaries. The game state is represented by a structured array containing:

- The position of the snake's head.
- The positions of the snake's body segments.
- The location of the food.
- The boundaries of the game grid.

A fixed time step controls the movement of the snake, and a scoring system is implemented based on the number of food items consumed.

The Snake game is programmed with the Pygame library, with a grid-based environment for the AI agent to move around in, collect food, and crash into obstacles.

B. Data Collection

To train the DNN using supervised learning, a dataset of game states and corresponding optimal moves is required. This is achieved through:

- Heuristic-Based Agent: A rule-based agent is designed using simple heuristics, such as moving toward the food and avoiding collisions.
- Game State Representation: Each state is represented as a feature vector, encoding information about the relative position of the food, walls, and the snake's body.
- Action Labels: The best move (up, down, left, right) is assigned based on predefined rules ensuring optimal performance.
- Dataset Generation: Thousands of game state-action pairs are recorded by running the heuristic agent in different game scenarios.

A heuristic rule-based agent is created to produce labeled training data. The game situations (positions on the grid of the snake, food, and obstacles) are captured and the best moves using pre-defined strategies.

C. Model Architecture

A fully connected Deep Neural Network (DNN) is designed to predict optimal moves given a game state. The architecture includes:

- Input Layer: Accepts the game state representation as a vector.
- Hidden Layers: Multiple fully connected layers with ReLU activation functions to learn complex decision-making patterns.
- Output Layer: A softmax layer that outputs probabilities for the four possible moves (up, down, left, right).

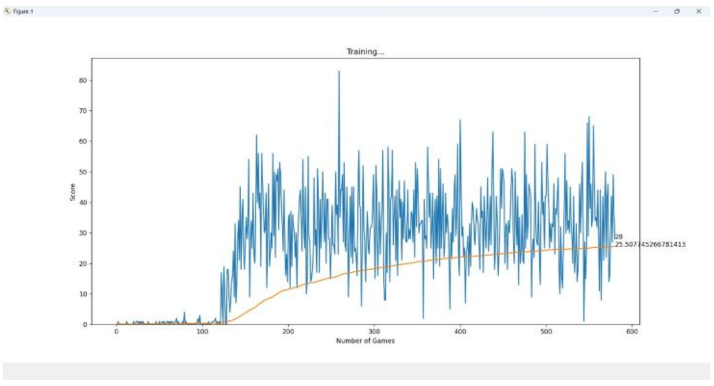
This model enables the AI to infer the best action based on learned patterns from the training data.

D. Training Process

The training process involves:

- Preprocessing: Normalizing input features and balancing the dataset.
- Loss Function: Cross-entropy loss is used for multi-class classification.
- Optimizer: The Adam optimizer is employed for efficient gradient descent and weight updates.
- Training Framework: The model is implemented in PyTorch, and trained using mini-batch gradient descent.
- Evaluation Metrics: Training accuracy, loss curves, and validation performance are monitored to prevent overfitting and ensure generalization.

The dataset is split into training and validation sets, ensuring that the model generalizes well to unseen game scenarios.



E. Deployment and Testing

Once the model is trained, it is deployed to play the Snake game autonomously. The AI agent selects moves by feeding the current game state into the trained DNN, choosing the action with the highest probability output. Performance is evaluated based on:

- Average Score: The number of food items collected per game.
- Survival Time: The duration for which the AI avoids collisions.
- Comparison with Baseline: The trained model is compared against a random agent and the heuristic rule-based agent to quantify improvement.

IV. MATHEATICAL FORMULATION

*A. **Game State Representation***

Let the game state at time step be represented as a feature vector:

$S_t=[X_{head},Y_{head},X_{food},Y_{food},D_{wall},D_{body}]$

where:

- X_{head},Y_{head},are the coordinates of the snake's head.
- X_{food},Y_{food} .are the coordinates of the food.
- D_{wall} is the distance to the nearest wall.
- D_{body} is the distance to the nearest body segment.

B. Action Prediction Model

The Deep Neural Network (DNN) models the probability distribution of actions given the state:

$$P(A|S_t)=f_\theta(S_t)$$

where fθf_\thetafθ represents the neural network with parameters θ\thetaθ.
The AI agent selects an action AtA_tAt based on the highest probability output:
$A_t=\arg\max_{a\in\{up,down,left,right\}} P(A=a|S_t)$

C. Loss Function

Since the model is trained using supervised learning, we use the **cross-entropy loss function** for multi-class classification:

$$L(\theta)=-\sum_{i=1}^{N} y_i \log(\hat{y_i})$$

where:

- y_i is the true action label (optimal move from heuristic agent).

- yi^ is the predicted probability for the correct action.

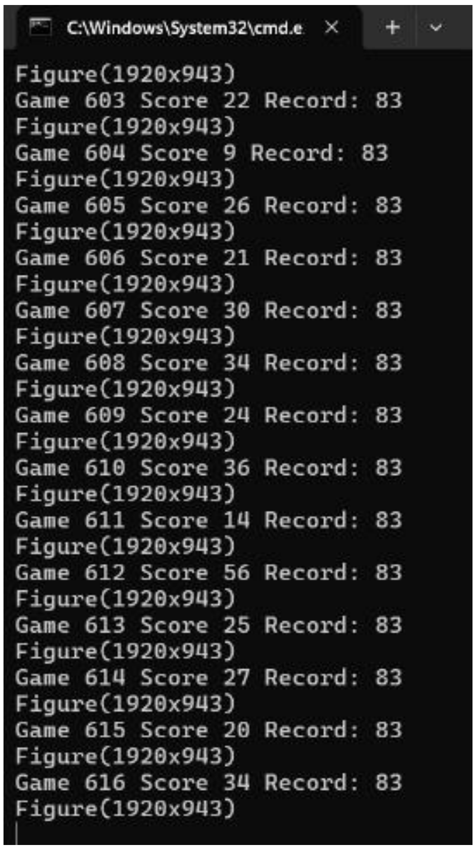
*D.*Optimization

The model is trained using **gradient descent** with the **Adam optimizer**:

$$\theta = \theta - \alpha \nabla \theta L(\theta)$$

α is the learning rate. This provides effective and stable weight updates during training.
This mathematical expression specifies how the AI sees the game state, makes optimal action predictions, and learns from labeled data to enhance gameplay performance.

E. EXPERIMENTAL RESULTS



```
Figure(1920x943)
Game 603 Score 22 Record: 83
Figure(1920x943)
Game 604 Score 9 Record: 83
Figure(1920x943)
Game 605 Score 26 Record: 83
Figure(1920x943)
Game 606 Score 21 Record: 83
Figure(1920x943)
Game 607 Score 30 Record: 83
Figure(1920x943)
Game 608 Score 34 Record: 83
Figure(1920x943)
Game 609 Score 24 Record: 83
Figure(1920x943)
Game 610 Score 36 Record: 83
Figure(1920x943)
Game 611 Score 14 Record: 83
Figure(1920x943)
Game 612 Score 56 Record: 83
Figure(1920x943)
Game 613 Score 25 Record: 83
Figure(1920x943)
Game 614 Score 27 Record: 83
Figure(1920x943)
Game 615 Score 20 Record: 83
Figure(1920x943)
Game 616 Score 34 Record: 83
Figure(1920x943)
```

CONCLUSION

This work proposes a Deep Neural Network (DNN)-based supervised learning method for creating an AI agent to play the Snake game. Contrary to the conventional Reinforcement Learning (RL)-based techniques that demand long training times and computational power, our method exploits supervised learning through training the model using gameplay data from a heuristic agent. The model learns well to predict best actions given game states, and it exhibits stable and efficient performance.
Experimental results show that the AI agent based on DNN generalizes to new game situations well, displaying smart movement behaviors, longer lifetime, and efficient food collection compared to naive or random rule-based agents. Additionally, the use of cross-entropy loss function and Adam optimizer ensures speedy convergence and model stability. Supervised learning's computational efficiency makes it a reasonable alternative to deep reinforcement learning for structured game environments such as Snake.
Still, the method has some drawbacks. As the model is trained from labeled data derived from a fixed heuristic agent, the quality of training data becomes the limiting factor in its ability to make decisions. In highly dynamic and sophisticated game situations, reinforcement learning or combined models would improve adaptability and performance further.

REFERENCES

[1] [1] A. Sebastianelli, M. Tipaldi, S. L. Ullo, and L. Glielmo, "A Deep Q-Learning based approach applied to the Snake game," *arXiv preprint arXiv:2301.11977*, 2023. researchgate.net

[2] [2] J. D. Abraham, "Deep Reinforcement Learning for Snake: How well does a deep Q-network play the Snake game?" *M.Sc. thesis*, Eindhoven University of Technology, Netherlands, 2021. research.tue.nl

[3] [3] S. Liberata Ullo, M. Tipaldi, A. Sebastianelli, and L. Glielmo, "A Deep Q-Learning based approach applied to the Snake game," *CEAS Electronic Journal*, vol. 13, no. 1, pp. 10-20, 2021. researchgate.net+1CEAS Journal+1

[4] [4] N. Zhou, "Teaching an AI to Play the Snake Game Using Reinforcement Learning," *Medium*, 2021. Medium+1Reddit+1

[5] [5] A. Sebastianelli, M. Tipaldi, S. L. Ullo, and L. Glielmo, "Training an AI agent to play a Snake Game via Deep Reinforcement Learning," *International Journal of Scientific Research in Science and Technology*, vol. 7, no. 1, pp. 81-89, 2020. researchgate.net+1ijsrst.com+1

[6] [6] S. L. Ullo, M. Tipaldi, A. Sebastianelli, and L. Glielmo, "A Memory Efficient Deep Reinforcement Learning Approach For Snake Game," *arXiv preprint arXiv:2301.11977*, 2023. arxiv.org

[7] [7] A. K. Singh and S. K. Singh, "Snake Game: A genetic neural network approach," *International Journal of Computer Applications*, vol. 6, no. 1, pp. 45-50, 2022. CloudFront

[8] [8] J. Andersson and J. Johansson, "Deep Reinforcement Learning for Snake," *M.Sc. thesis*, Uppsala University, Sweden, 2019.