AI Snake

Kamil Kozioł, Magdalena Sadowska, Wiktor Leszczyński

1 Introduction

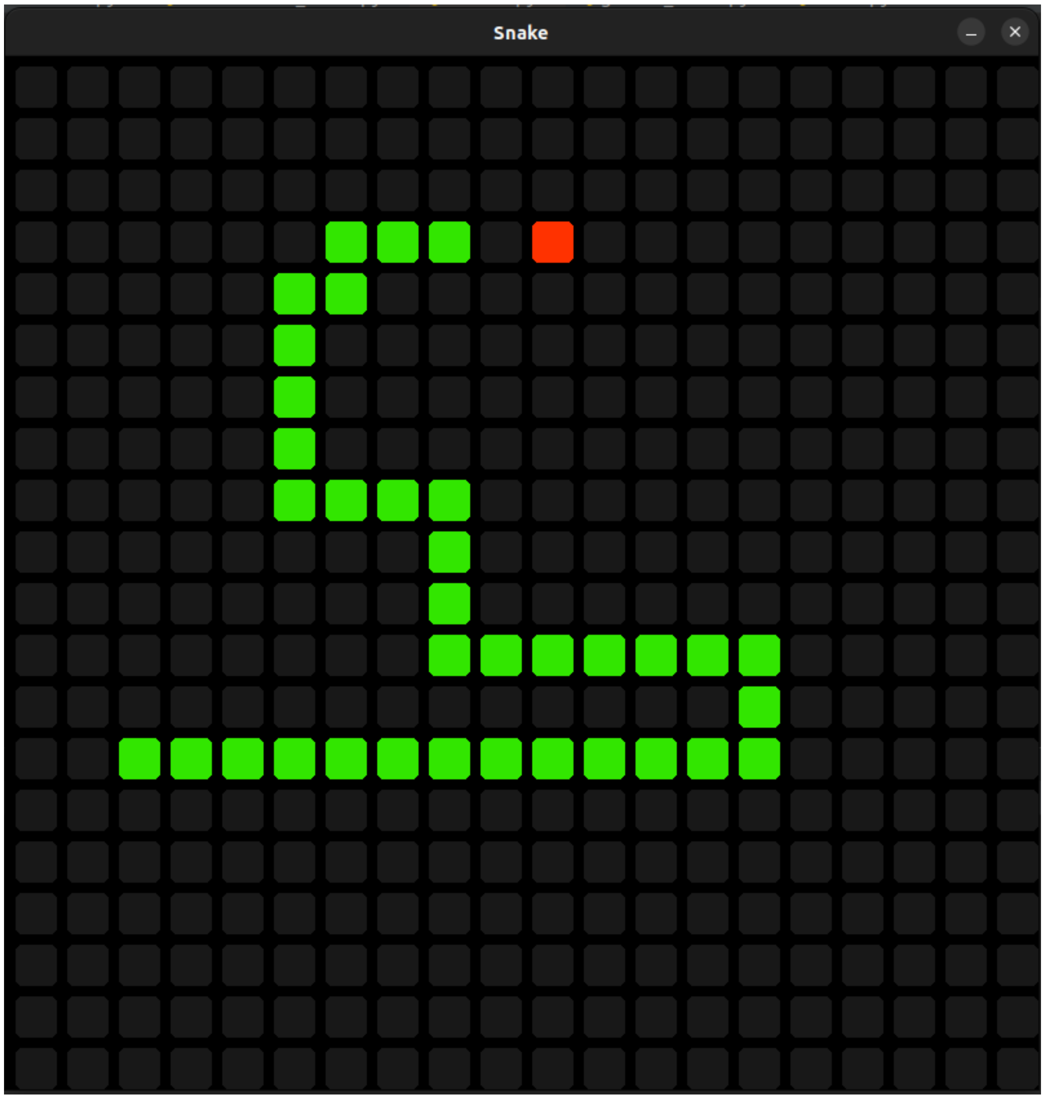
This project intends to compare 4 different AI implementations which can be used when making bots that play games

This project intends to compare and implement four different AI implementations that can be utilized for creating game-playing bots. The focus will be on comparing the performance and effectiveness of :

- Genetic Algorithm,
- upervised Classification,
- Deep Q-Network (DeepQ),
- NeuroEvolution of Augmenting Topologies (NEAT),

2 Game

The game in which the algorithms will be tested is an original implementation of the well-known game Snake. This custom Snake game is developed using the Pygame library, which provides a set of tools and functionalities for creating 2D games.



3 Genetic algorithm

Genetic Algorithm is a machine learning technique that uses principles of natural selection to find the optimal solution to a problem.

We can determine how good individual is doing by calculating his fitness.

3.1 Algorithm

Each of these steps is repeated untill conditions are met and then the current generation value is increased

3.1.1 Initialization

The first step is initializing a population using random neural networks, where each solution represents a possible strategy for playing the game.

Done only once at the first generation

3.1.2 Evaluation

Each solution is evaluated by playing the game using the corresponding neural network. Fitness score of the solution is calculated based on its performance in the game. The fitness score is a measure of how well the solution performs in achieving the objective of the game.

3.1.3 Selection

Then propabilities of picking each individual are made. Those propabilities range from $\langle 0, 1 \rangle$

3.1.4 Mutation

To introduce diversity into the population, picked by random individuals undergo mutation, where random changes are made to their weights and biases. This process helps to prevent the population from converging too quickly to a local optimum.

Every weight and bias has a change of being mutated based on $mutationRate$

If mutated those are modified using this formula:

$$x = x + randomGaussian(initMean, initStdev) * mutationPower$$

Then being kept at $\langle minValue, maxValue \rangle$

3.1.5 Crossover

The selected solutions are then combined through crossover, where random pairs of individuals exchange weights and biases of neural network to create new offspring solutions. This process mimics the natural process of sexual reproduction, where genes from two parents combine to produce offspring with a mix of genetic traits.

3.1.6 Elites

Some of the best-performing solutions from the previous generation are also added to the new generation

3.2 Fitness function

Fitness function is calculated by this formula:

$$fitness = \begin{cases} age^2 * (2^{apples}) & apples < 10 \\ age^2 * (2^{10}) * (apples - 9) & apples \geq 10 \end{cases}$$

Each snake in the game has a hunger variable, denoted as H. Upon consumption of an apple, H is replenished to a maximum value of $H_{\max}$. At every time step, H decreases and once it reaches 0, the snake perishes.

3.3 Hyperparameters

Hyperparameters can be modified in file *settings.json*.

3.3.1 Neural network

Neural network used to train the snakes is described by this model:

$$Model = \begin{matrix} InputLayer(28, linear) \\ DenseLayer(20, relu) \\ DenseLayer(12, relu) \\ OutputLayer(4, softmax) \end{matrix}$$

Description:

$$Layer(neurons, activationFunction)$$

4 NEAT

We have now added a title, author and date to our first LaTeX document!

4.1 Podsekcja

ziut

5 DeepQ

We have now added a title, author and date to our first LaTeX document!

5.1 Podsekcja

ziut

6 Supervised Classification

The objective of this report is to outline implementation of supervised classification methods for the game Snake. By using a dataset of pre-labeled game states and employing machine learning algorithms, aim was to create a snake-playing agent capable of making informed decisions to achieve higher scores. This report presents the methodology, results, and analysis of our approach, highlighting the effectiveness and potential of supervised classification in the context of gaming applications.

6.1 Implementation

In the project, the TensorFlow library was used to implement supervised classification

6.2 Data

The dataset used in this project consists of gameplay data recorded from our own gameplay sessions. By capturing and labeling the game states during our playthroughs of Snake, we created a dataset specifically suited for our implementation. This approach allows us to train our supervised classification models using real gameplay scenarios. Leveraging our own gameplay data adds an element of authenticity and enables us to develop a snake-playing agent that can perform well in similar gameplay conditions.

The dataset used in this project comprises 28 parameters, specifically rays that measure distances. These parameters are categorized into four groups: the first eight represent distances from walls, the next eight represent distances from the apple, the next eight are distances to the snake's own body, and the remaining four values indicate the direction of movement.

6.3 Neural network

Neural network used to train the snakes is described by this model:

$$Model = \begin{array}{l} InputLayer(28, linear) \\ DenseLayer(28, relu) \\ DenseLayer(16, relu) \\ OutputLayer(4, softmax) \end{array}$$

6.4 Performance

The effectiveness of the agent is primarily influenced by two factors: the quantity of data available for training and the skills of the players on whose gameplay sessions the agent was trained. A larger dataset provides the agent with a wider range of scenarios to learn from, enabling it to make more accurate decisions. Additionally, the agent's performance is closely linked to the skills and strategies exhibited by the players whose gameplay sessions were utilized for training, as the agent learns from their patterns and behaviors. When players did not use any specialised techniques, or patterns, the bahaviour of the agent seemed to be a bit chaotic.

6.5 Summary

In conclusion, the implementation of supervised classification techniques using the TensorFlow library has proven effective in enhancing the gameplay of the Snake game. By leveraging a dataset consisting of labeled gameplay data, the agent demonstrates the ability to make decisions, resulting in improved performance and higher scores. Furthermore, utilizing player-based data contributes to a more natural and human-like pattern of movement, distinguishing it from other learning methods like genetic algorithms. However, the main limitation lies in the quantity and quality of the training data. Incorporating data from various players with different playing styles can lead to suboptimal decisions by the agent.

7 Summary

This project intends to compare 4 different AI implementations which can be used when making bots that play games

7.1 Podsekcja

ziut DAWAJ TEKST