Snake Game AI Using Supervised Learning

Harsh Tripathi, Navdeep Singh, Pankaj

November 18, 2018

1 Problem Statement

Creating a snake game agent that plays the game on it's own using supervised learning approach, the problem is usually solved by reinforcement learning.

2 Introduction

The game is classic nokia mobile snake game in which a snake eats and increases it's size by one block after eating the fruit, the snake dies if it touches the boundaries or itself. The problem is to find the safe path from the head of the snake to the fruit. The fruit decomposes after some amount of time and then grows at some other random location when not eaten or grows again somewhere else on the grid once eaten. Dataset is created by playing the game created during the process and cleaned for inconsistency. Three models are trained Logistic Regression, Random Forest and XGBoost Classifier, out of which XGBoost performed better.

3 About the Dataset

Up : It contains 1 if path is blocked ahead of snake.

Down : It contains 1 if path is blocked below the snake.

Left : It contains 1 if path is blocked to the left of snake.

Right : It contains 1 if path is blocked to the right of snake.

Angle : It tells the orthogonal angle that fruit makes with the head of the snake in radians.

Distance : Euclidean distance between snake head and fruit.

Decision : It contains -3,+3,-1,+1 denoting left,right,up,down.

4 Data Collection

Data Collection is done by user playing the created game to generate about 7000 rows of data which took about one hour. To save the data a file is created and each time the state of the game is changed up, down, left, right, angle, distance, decision taken by the user are calculated and stored as a row in the file. The game restarts on it's own if the snake dies.

5 Approach

User plays game for some time and generates data. This data then is cleaned for wrong decisions taken by the user that played it. After this a model is created and tested. Then the model is saved into a file using pickle library.

On the first attempt, we created Logistic Regression with 1000 iterations in training, which gave the accuracy score of about 0.786 in predicting the correct move.

On the second attempt, we created Random Forest with 1000 estimators, which increased the accuracy score to 0.833.

On the third attempt, we created XGBoost Classifier with 300 estimators and depth of 8, which increased the accuracy score to 0.858.

6 Feature Engineering

In this we dropped all rows for which the decision taken by the user while playing game to generate the data was wrong. For example if the left direction is blocked and snake still moves left then the row must obviously be dropped like this about 150 rows were dropped. Which improved the accuracy of models a lot, especially XGBOOST.

7 Results

We used accuracy score to measure the differences between values predicted by our model and the values observed. Out of total test cases of 1821, number of true positive were 1517 for XGBOOST.

The approach taken i.e supervised learning does not take into consideration that the snake should definitely find the path to fruit at all time, and the approach taken has no way to tackle this problem of forcing the snake to find the path to the fruit.

Therefore sometimes the snake moves in a cycle and to break the cycle of that movement, the change of position of fruit, when it decomposes after some amount of time helped a lot.