Semester VI Mini Project

Maze Solving Using Machine Learning Techniques

L.Aniruth Naraayanan
Siddesh Kokane

Domain:

Machine Learning

Keywords:

Machine Learning, Maze Solving, Deep Learning, Q-training/testing, Reinforced Learning

Introduction:

In this Project we aim to solve the classical Maze Problem using modern techniques such as deep learning and reinforced learning. We, also aim to provide additional solutions for various scenarios which will be benefited by machine leaning techniques.

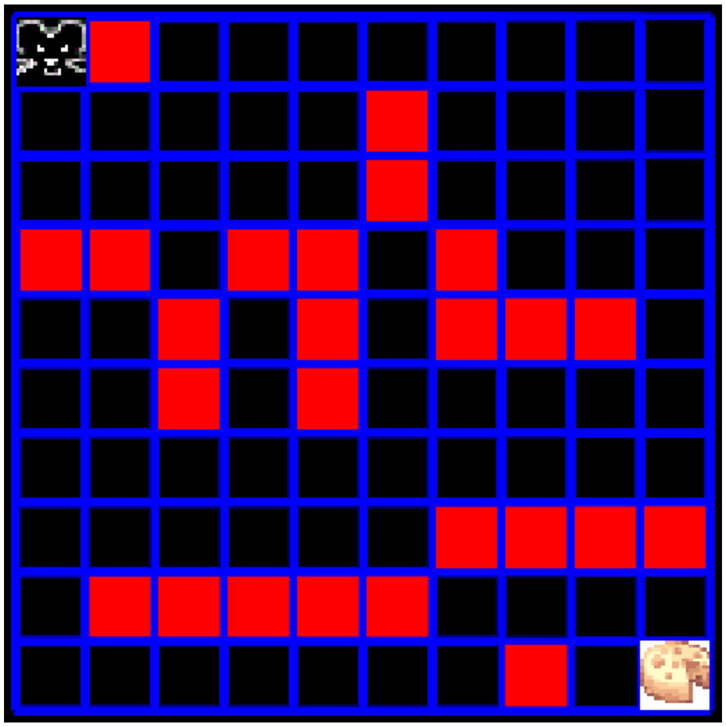
Reinforced Learning:

Reinforcement learning is a machine learning technique for solving problems by a feedback system (rewards and penalties) applied on an agent which operates in an environment andneeds to move through a series of states in order to reach a pre-defined final state.
A classical example is a rat (agent) which is trying to find the shortest route from a starting cell to a target cheese cell in a maze (environment). The agent is experimenting and exploiting past experiences (episodes) in order to achieve its goal. It may fail again and again, but hopefully, after lots of trial and error (rewards and penalties) it will arrive to the solution of the problem.

The solution will be reached if the agent finds the optimal sequence of states in which the accumulated sum of rewards is maximal (in short, we lure the agent to accumulate a maximal reward, and while doing so, he actually solves our problem).
Note that it may happen that in order to reach the goal, the agent will have to endure many penalties (negative rewards)on its way. For example, the rat in the above maze gets a small penalty for every legal move.
The reason for that is that we want it to get to the target cell in the shortest possible path. However, the shortest path to the target cheese cell is sometimes long and winding, and our agent (the rat) may have to endure many penalties until he gets to the "cheese" (sometimes
called "delayed reward").

Maze Solving:

Traditional maze puzzles have been used a lot in data structures and algorithms research and education. The well-known **Dijkstra shortest path** algorithm is still the most practical method for solving such puzzles, but due to their familiarity and intuitive nature, these puzzles are quite good for demonstrating and testing Reinforcement Learning techniques.
A simple maze consists of a rectangular grid of cells (usually square), a rat, and a "cheese"(target).



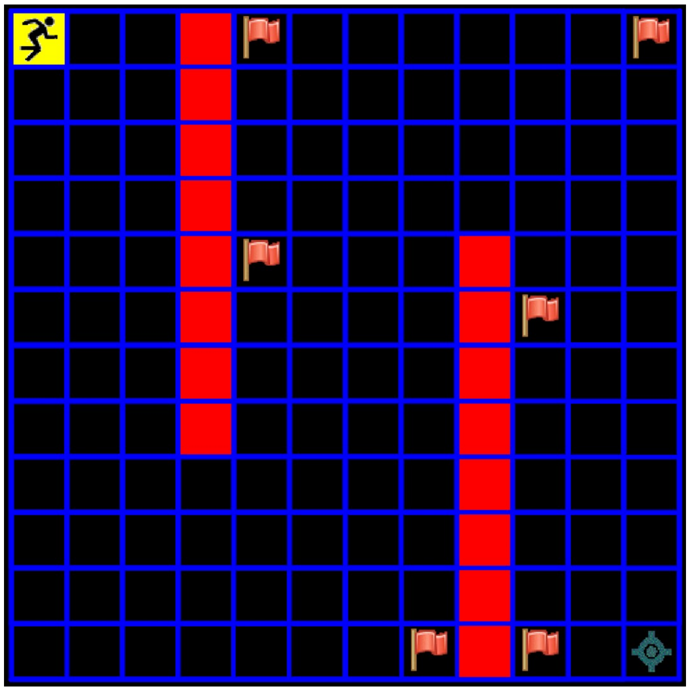
Extended Solutions for other scenario for Course Projects:

1.A more challenging maze problem is a cat/mouse chase puzzle in which the blocked cells, as well as the **mouse are moving in time**. This is the type of problems in which classical algorithms are starting to get rough and deep learning techniques could provide better answers.

2. There are plenty of variations on the above type of games. We can for example add a cheese to the maze and then we have a double chase scene: the cat chases the mouse, and the mouse chases the moving cheese.
This is also known as the **Theseus and Minotaur Maze** and it belongs to the multi-agent type of problems.

3. We can throw **several pieces of cheese** to the maze and the rat will have to find the shortest route for collecting all of them. Of course, we can also add complications like moving cells and moving cheese, but it seems like the static version is hard enough.

4. In our next notebook we will explore the **"Tour De Flags"** maze in which an agent has to collect a group of flags and deliver them to a target cell. The agent must find a shortest route for doing so. The agent receives bonus points for collecting each flag, and receives the full award when he arrives to the target cell.



Technology Stack:

Programming Language used:	**Python**-3
Back-end ML Framework:	**Tenserflow** with virtualenv
Neural Network Framework:	**Keras** library
GUI:	**Tkinter** and matplotlib(for platting cells)

Conclusion:

We hope to complete this project successfully with all our objectives fulfilled.
We sincerely think this project will help us gain some valuable knowledge and experience in the ever-evolving domain of machine learning.

References:

https://www.samyzaf.com/ML/rl/qmaze.html

Demystifying Deep Reinforcement Learning
(https://www.nervanasys.com/demystifying-deep-reinforcement-learning)

Keras plays catch, a single file Reinforcement Learning example
(http://edersantana.github.io/articles/keras_rl)

Q-learning (https://en.wikipedia.org/wiki/Q-learning)

Keras plays catch - code example
(https://gist.github.com/EderSantana/c7222daa328f0e885093)

Reinforcement learning using chaotic exploration in maze world
(http://ieeexplore.ieee.org/document/1491636)

A Reinforcement Learning Approach Involving a Shortest Path Finding Algorithm
(http://incorl.hanyang.ac.kr/xe/paper/ic/ic2003-3.pdf)

Neural Combinatorial Optimization with Reinforcement Learning
(https://arxiv.org/abs/1611.09940)

Google Acquires Artificial Intelligence Startup DeepMind For More Than $500M (https://techcrunch.com/2014/01/26/google-deepmind)

How DeepMind’s Memory Trick Helps AI Learn Faster (https://www.technologyreview.com/s/603868/how-deepminds-memory-trick-helps-ai-learn-faster/?imm_mid=0ef03f&cmp=em-data-na-na-newsltr_20170320)

Methods and apparatus for reinforcement learning US 20150100530 A1

Intro to Reinforced Learning(http://www.cs.indiana.edu/~gasser/Salsa/rl.html)

Reward function and initial values: Better choices for accelerated Goal-directed reinforcement learning (https://hal.archives-ouvertes.fr/hal-00331752/document)

Andrej Karpathi blog: Deep Reinforcement Learning: Pong from Pixels (http://karpathy.github.io/2016/05/31/rl/)

Thank You