

BitByBit

LAB-8

Matchbox Educable Naughts and Crosses Engine

TEAM MEMBERS :

AYUSHI SHUKLA - 202051044

BHARGAVI KAMBLE - 202051048

DEVASHISH ANIL DESHMUKH -202051061

MANIYAR AAMER SOHEL - 202051113



CONTENT OF THIS PRESENTATION



PROBLEM STATEMENT

Explaining The Problem Statement

MDP and RL

Basics of data structure needed for state-space search tasks and use of random numbers required for MDP and RL

MENACE

Explaining What is MENACE and its history.

Why Matchbox machine?

MENACE Strategies and how it learns.

Implementation of MENACE Machine Learning Model and the strategies Used.

Observations,Results and Conclusion

Observations , Results and conclusion after performing this experiment/lab.



PROBLEM STATEMENT

Read the reference on MENACE by Michie and check for its implementations. Pick the one that you like the most and go through the code carefully. Highlight the parts that you feel are crucial. If possible, try to code the MENACE in any programming language of your liking.

MENACE

MENACE - Machine Educable Noughts And Crosses Engine

It is an early example of a machine learning algorithm developed by Donald Michie in the 1960s.

The algorithm was designed to play the game of noughts and crosses, also known as tic-tac-toe, and learn from its mistakes through a process of trial and error.



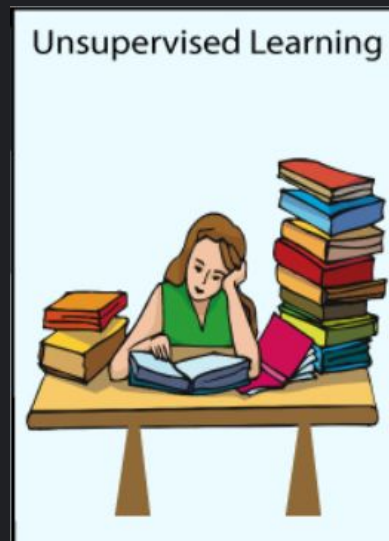
THEORY & BACKGROUND OF MENACE



WHAT WE ARE GOING TO DISCUSS TODAY ?



- > Spoon feeding the machine
- > Dataset has instances with input and corresponding output.



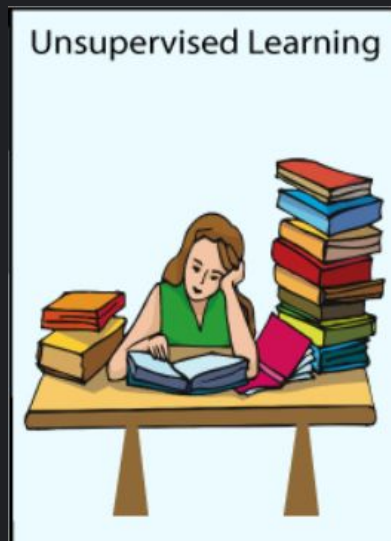
- > Let machine learn from its mistakes and its achievements
- > Learn like HUMANS



WHAT WE ARE GOING TO DISCUSS TODAY ?



- > Spoon feeding the machine
- > Dataset has instances with input and corresponding output.



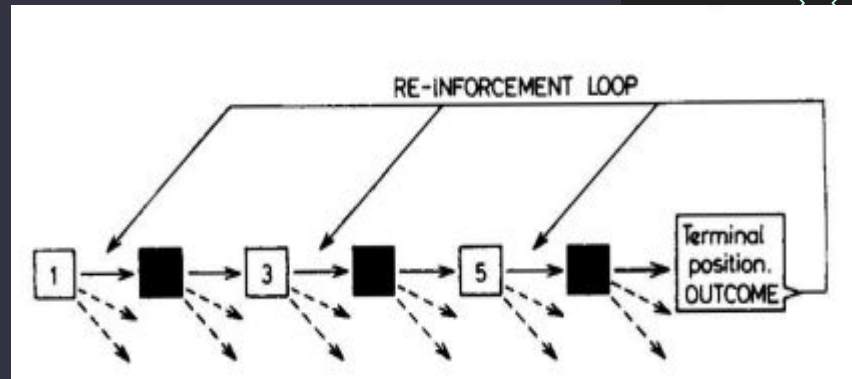
- > Let machine learn from its mistakes and its achievements
- > Learn like HUMANS



REINFORCEMENT LEARNING (RL)

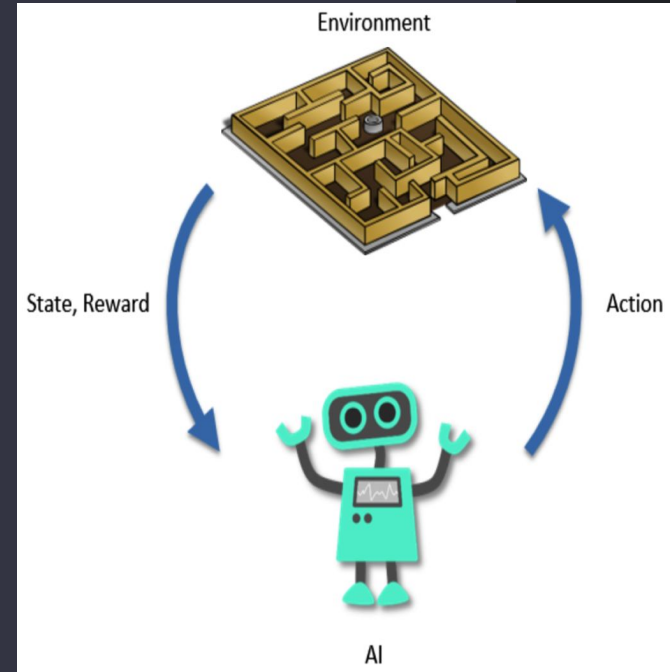
Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions.

- For each good action, the agent gets positive feedback.
- For each bad action, the agent gets negative feedback or penalty.
- It is based on the hit and trial process.
- The environment has random probability distribution, and the agent needs to explore it to reach to get the maximum positive rewards.



MARKOV DECISION PROCESS (MDP)

- MDP, is used to formalize the reinforcement learning problems. If the environment is completely observable, then its dynamic can be modeled as a Markov Process.
- In an MDP,
 - > The agent interacts with an environment in a sequence of discrete time steps.
 - > At each time step, the agent observes the current state of the environment and selects an action to take.
 - > The environment then transitions to a new state, and the agent receives a reward or penalty based on the action taken and the new state of the environment.
 - > The goal of the agent is to learn a policy that maximizes its long-term cumulative reward.



MARKOV DECISION PROCESS (MDP)

The reinforcement learning problem is typically modeled using Markov Decision Processes. A Markov decision process (MDP) is defined by a tuple of four entities (S, A, T, r)

where,

S - state space,

A - action space,

T - transition function that encodes the transition probabilities of the MDP

r - is the immediate reward obtained by taking action at a particular state.

Consider a trajectory sequence,

$$\tau = (s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, \dots).$$

Then total reward obtained of above trajectory is

$$R(\tau) = r_0 + r_1 + r_2 + \dots.$$

MARKOV DECISION PROCESS (MDP)

$$R(\tau) = r_0 + \gamma r_1 + \gamma^2 r_2 + \dots = \sum_{t=0}^{\infty} \gamma^t r_t.$$

discount factor $\rightarrow \gamma < 1$


To balance between Exploration VS Giving Quick - Solution

Here note that:


- if γ is very small, the rewards earned by the robot in the far future, say $t=1000$, are heavily discounted by the factor γ^{1000} . This encourages the robot to select short trajectories that achieve its goal, namely that of going to the goal in the environment.
- But if $\gamma=0.99$, the robot is encouraged to explore more and then find the best trajectory to go to the goal location.



RANDOM NUMBERS REQUIRED FOR MDP & RL



Random numbers are used to represent the uncertainty and stochasticity in problem and environment




In MDP :


Transition probabilities (T):

- In an MDP, the transition probabilities specify the probability of moving from one state to another when an action is taken.
- Since the outcome of an action is uncertain, random numbers are used to represent the stochasticity associated with the transition probabilities.

Rewards (r):

- 
- In an MDP, the rewards specify the immediate reward or penalty associated with taking an action in a particular state.
 - Since the rewards may be stochastic, random numbers are used to represent the uncertainty associated with the rewards.

Example :

- 
- For example, in a game of poker, the cards dealt to each player are random, and the outcome of the game depends on the cards each player receives. In MDPs, random numbers are used to represent this uncertainty and to simulate the possible outcomes of actions.

RANDOM NUMBERS REQUIRED FOR MDP & RL

In RL:

- Random numbers are often used in exploration strategies.
- Exploration is the process of trying out different actions to learn more about the environment and to discover the best actions to take in the long run.
- Since RL algorithms aim to learn optimal policies by maximizing rewards over time, it is crucial to explore the environment to find the best actions.
- Random numbers are often used to introduce randomness into the agent's actions during exploration, which helps the agent to explore new states and learn about the environment.

MENACE- The Computer Made From Matchboxes

MENACE stands for **Machine Educable Noughts and Crosses Engine** [1]. It was originally described by **Donald Michie**, who used **304 matchboxes** to record each game he played against this algorithm.

This provides an adequate conceptual basis for a trial-and-error learning device, provided that the total number of choice-points which can be encountered is small enough for them to be individually listed. Michie's aim was to prove that a computer could **"learn"** from failure and success to become good at a task.





Need For Matchbox Machine



Matchboxes were used because each box contained an assortment of variously coloured beads. The different colours correspond to the different unoccupied squares to which moves could be made.





Fig[1]. MENACE originally created by Donald Michie

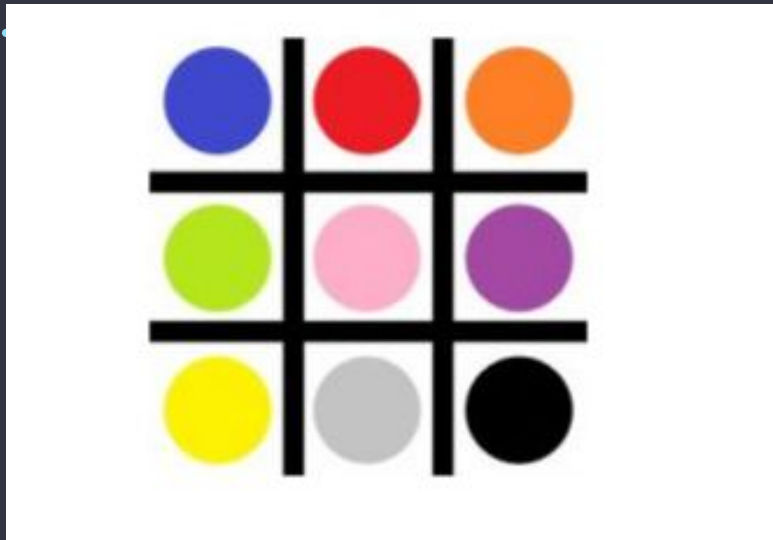


Fig. 2: The colour code used in the matchbox machine.



The system of numbering the squares is that adopted for the subsequent computer simulation program

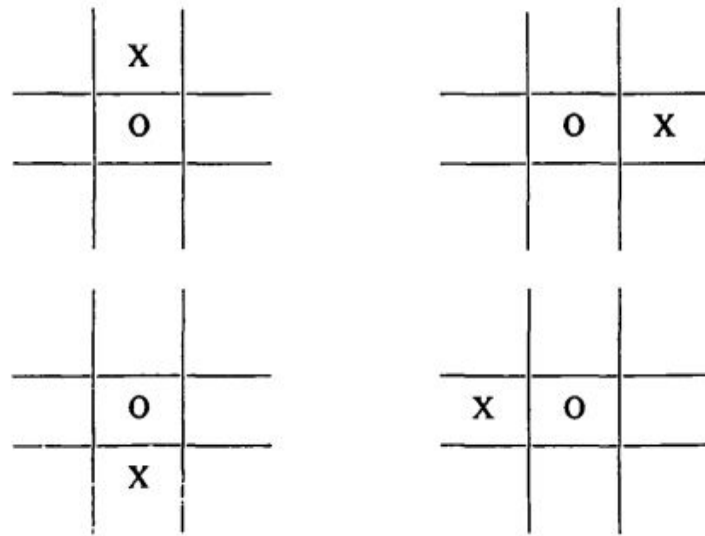


Fig. 3.-Four positions which are in reality variants of a single position

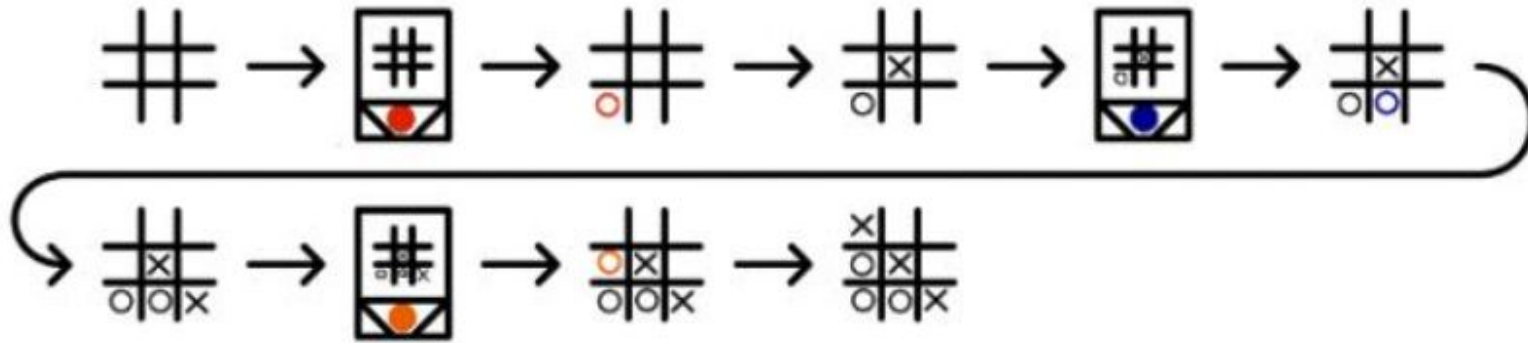


MENACE Strategy and The working of the Learning Model

MENACE (Machine Educable Noughts And Crosses Engine) is a learning model used to teach a computer to play noughts and crosses. The model is based on a reinforcement learning approach and uses a series of plastic beads to represent the possible moves available to the computer player. These beads are distributed amongst a series of boxes, corresponding to the different possible board configurations. When the computer moves, it picks a box with beads in it at random and selects a move corresponding to the bead's position in the box. **If the move leads to a win, the beads in that box are rewarded by adding additional beads of the same color.** If the move leads to a loss, the beads are punished by removing beads of that color. With enough training, the beads in each box come to represent the optimal move for that board configuration, and the computer becomes unbeatable.



AN EXAMPLE :



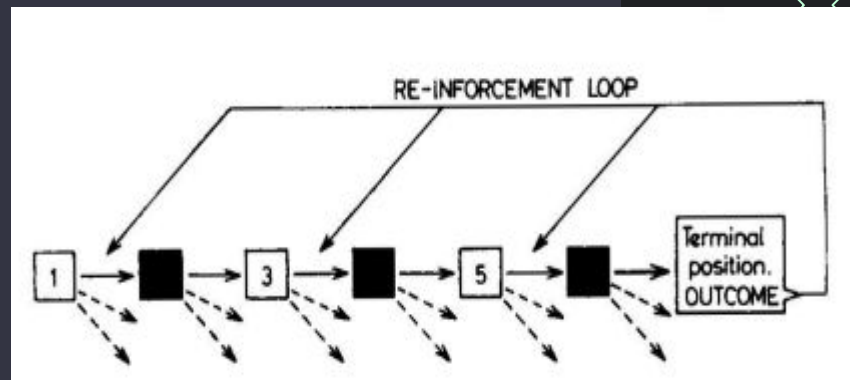
- MENACE **lost the game above**, so the beads that were chosen are **removed from the boxes**. This means that MENACE will be **less likely to pick the same colours again** and has learned.
- If MENACE had **won**, **three beads of the chosen colour would have been added to each box**, encouraging MENACE **to do the same again**.
- If a game is a **draw**, **one bead is added to each box**.



REINFORCEMENT LEARNING (RL)

Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions.

- For each good action, the agent gets positive feedback.
- For each bad action, the agent gets negative feedback or penalty.
- It is based on the hit and trial process.
- The environment has random probability distribution, and the agent needs to explore it to reach to get the maximum positive rewards.



RANDOM NUMBERS IN MENACE

- If reward is increased :
In code implementation , we specified reward=3, penalty = -1;

What if reward is increased to reward=10, penalty=-1;

Then, Machine is less likely to explore a sub-branches of negative feedback path.
- If beats in each matchbox is increased ,
In code implementation, we specified that in each matchbox there are 2 beads for each move ;

What if nos of beads =2 is increased to 10;

Then, Machine is more likely to explore a sub-branches of negative feedback path




CODE AND WORKING OF THE MENACE MACHINE

Following is the link for [MENACE Machine](#) code explaining all the functions and observations of the learning Model


OBSERVATIONS



- The Menace game demonstrated the power of reinforcement learning in training a machine to play a simple game like tic-tac-toe without explicit programming of the rules. 
- Menace used a simple neural network consisting of matchboxes filled with colored beads to represent different states of the game and the moves that Menace could make. The beads were used to encode the machine's learning and decision-making process.
- Menace was trained through a process of trial and error, where the machine played against itself and learned from its mistakes. The reinforcement signal was provided by the beads in the matchbox

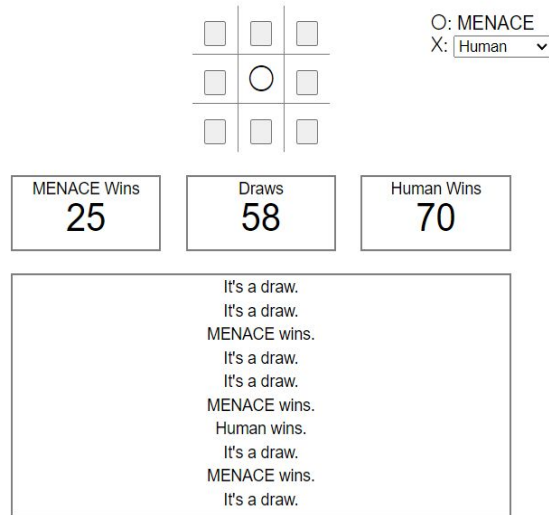
OBSERVATIONS



- Menace is an example of a **simple form of artificial intelligence** that can learn from its environment and make decisions based on that learning. It paved the way for more advanced machine learning algorithms and artificial intelligence systems that can learn and make decisions in more **complex and dynamic environments**. 

- To reduce the number of matchboxes required to build it, MENACE always plays first. (304 **matchboxes** if played this way, otherwise sampling will require **additional 470 matchboxes!**)

OBSERVATIONS



A sample MENACE GAME . After playing nearly 110 games , It becomes difficult to beat computer it is either a tie or MENACE's win. The following game can be found [here](#).



OBSERVATIONS



```
 0 | 0 | X
---+---+---
 0 | 0 | X
---+---+---
 X |   | X
```

MENACE moved : 5

Process finished with exit code 0

TRAINED MENACE

```
MENACE moved : 4
Enter your move : 2
```

```
 0 | 0 | 0
---+---+---
 X | X |
---+---+---
  |  |
```

Process finished with exit code 0

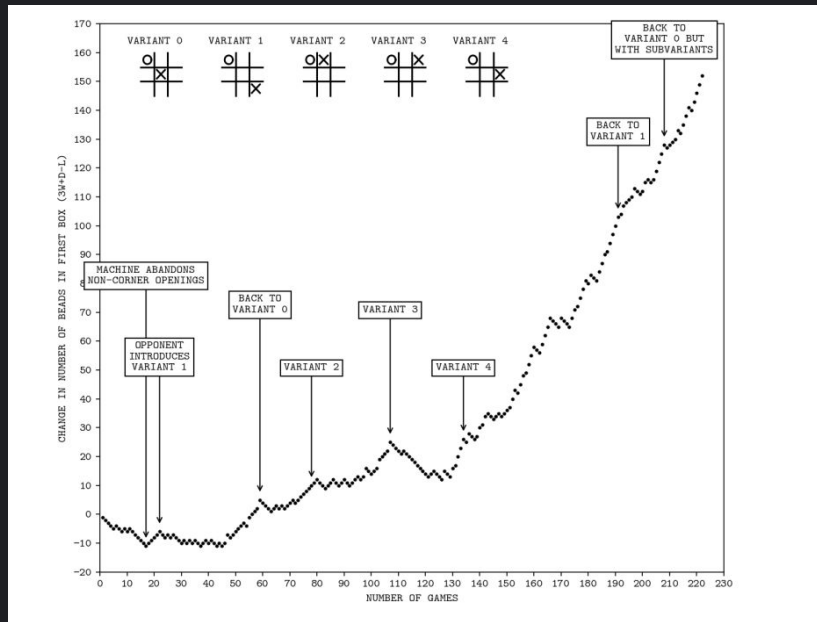
UNTRAINED MENACE



NOTE- X represents machine moves here and nought represent Player's move!

RESULTS

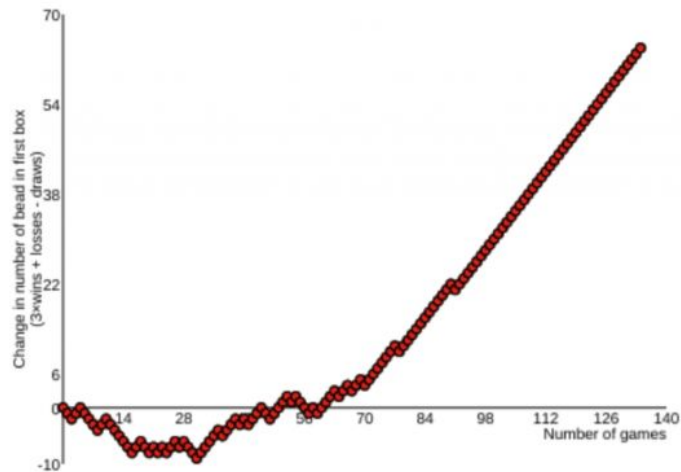
In Donald Michie's original tournament against MENACE, which lasted **220 games** and 16 hours, MENACE **drew consistently** after 20 games.



Fact : If player 1 and player 2 are playing game then if player 1 takes first turn then winning probability of player 1 : player = 2:1

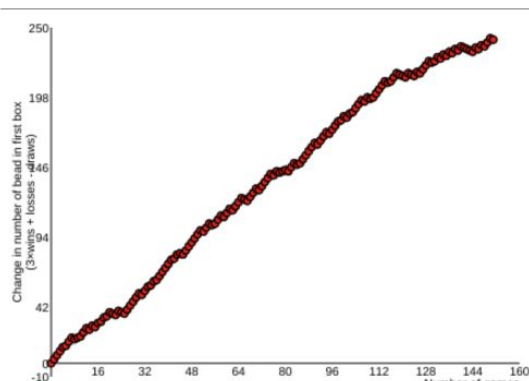
The progress of MENACE'S maiden tournament against a human opponent. The line of dots drops one level for a defeat, rises one level for a draw and rises three levels for a victory. Refer the following Figure.

TRAINED MENACE



When MENACE played a with a random picking opponent, the result is a near-perfect positive correlation

When MENACE plays a perfect-playing computer, the results look like this:
The Red colour symbolises that most of the games were draw!



REFERENCES

- https://d2l.ai/chapter_reinforcement-learning/mdp.html#return-and-discount-factor
- <http://people.csail.mit.edu/brooks/idoocs/matchbox.pdf>
- <https://www.msccroggs.co.uk/blog/19>
- <https://www.msccroggs.co.uk/menace/>
- <https://www.msccroggs.co.uk/blog/94>
- <https://github.com/thepushkarp/cs362-naagraaj>

The background is a dark navy blue. A large, dark grey rectangular block is centered horizontally and vertically, serving as a backdrop for the text. To the left of this block, there is a light green semi-circular shape at the top and a light blue circle below it. To the right, there is a light blue semi-circular shape at the bottom and a white outline of an 'X' above it. In the top right corner, there is a grid of 20 small white dots arranged in 4 rows and 5 columns. In the bottom left corner, there is a grid of 20 small white dots arranged in 4 rows and 5 columns.

THANK YOU