

Northeastern



Tic Tac Toe using Reinforcement Learning

Objective: To play tic tac toe with computer.

To understand how machine understands a tic tac toe problem and learns from it.

Problem Type: Reinforcement Learning Classification

Other problems:

https://www.youtube.com/watch?v=chrzpnL1OEM

Reinforcement Learning:

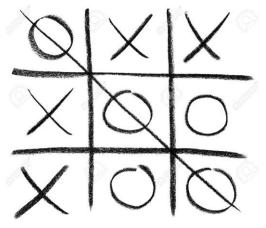
Ideal, human learning occurs either due to one of these,

- Rewards achieved when a task is completed.
- Fear of repercussions when task is incomplete.

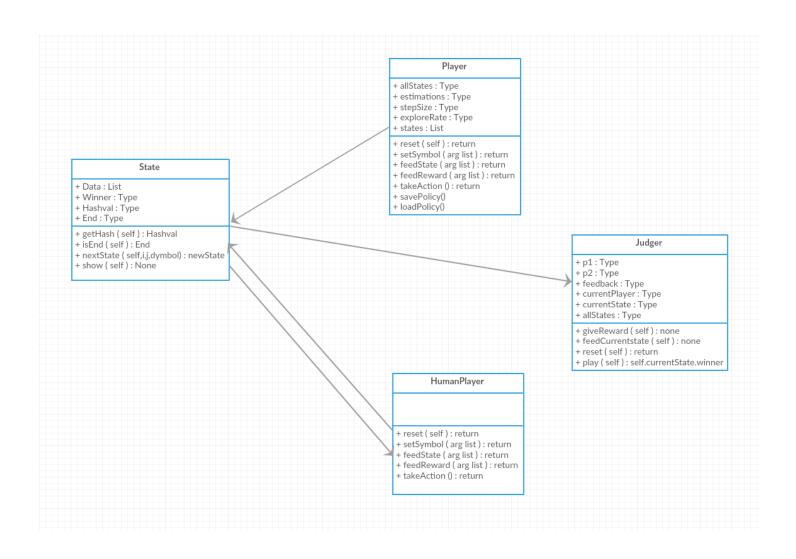
Reinforcement learning occurs when a machine is trained to complete a task with no pre-training steps and yes, assured of a reward for task completion.

T (S,a,S`) - Training and R(S,a,S`) - Rewards

Tic Tac Toe problem:



Tic Tac Toe problem – Class diagram



Class - State

State

- + Data : List
- + Winner: Type
- + Hashval: Type
- + End : Type
- + getHash (self): Hashval
- + isEnd (self): End
- + nextState (self,i,j,dymbol) : newState
- + show (self): None

What's the purpose of class State?

- State creates data, winner, hashval, end
- data is used to create 3x3 matrix array with all 0s
- Matrix takes values -1,0,1
- If 1, then * and if -1, then x else 0
- getHash() gives unique value for all 9 boxes
- isEnd() tie, win, lose condition check
- nextState(i,j,symbol) puts symbol in specified position
- show() Displays board

Class - Judger

Judger

- + p1 : Type
- + p2 : Type
- + feedback: Type
- + currentPlayer: Type
- + currentState : Type
- + allStates : Type
- + giveReward (self): none
- + feedCurrentstate (self): none
- + reset (self): return
- + play (self): self.currentState.winner

What's the purpose of class Judger?

- Judger- Initialize players, symbols, rewards and states
- Two players intialization
- Assign and set symbols for each
- Initializing states
- giveReward() Feedreward for the winning player
- feedCurrentState() feed moves of each players
- play() get moves from either players and save it to current state

Class – Al Player

Player

- + allStates : Type
- + estimations : Type
- + stepSize : Type
- + exploreRate: Type
- + states: List
- + reset (self): return
- + setSymbol (arg list) : return
- + feedState (arg list) : return
- + feedReward (arg list) : return
- + takeAction (): return
- + savePolicy()
- + loadPolicy()

What's the purpose of class Player?

- Player Initialize estimation, stepsize and explorerate
- setSymbol() save estimations using hashval for future runs. Estimation set to 1 and saved if winner. Else set to 0 and saved if loses.
- takeAction Using estimations and exploreRate values, take appropriate position to move to next state.
- savePolicy() saves estimation
- loadPolicy() loads all saved estimation

Class – HumanPlayer

HumanPlayer

- + reset (self) : return
- + setSymbol (arg list) : return
- + feedState (arg list) : return
- + feedReward (arg list) : return
- + takeAction (): return

What's the purpose of HumanPlayer?

- Initializes symbol and currentState
- takeAction() moves to appropriate position