Artificial Intelligence Based Videogame To Treat Patients With Schizophrenia

K.Umamaheswari[1]and G.Sindhu[2]
{hod.it@psgtech.ac.in[1], gsu.it@psgtech.ac.in[2] }

Professor, Information Technology,PSG College of Technology, Coimbatore[1], Assistant Professor, Information Technology,PSG College of Technology, Coimbatore[2].

Abstract. Machine learning algorithms can be broadly classified into supervised, unsupervised and reinforcement learning models. The well-suited technique for video game using machine learning is reinforcement learning. Reinforcement learning algorithm is about taking suitable action to maximize reward in any particular situation. Here there is no label, the agent decides what to do to perform given any task. Training process is not required. Patients with schizophrenia tend to experience an array of symptoms that affect their mental health. They do not co-operate with the treatment and loose hope quite often and very fast. Q learning is a reinforcement-based algorithm. The main advantage is that it is model free algorithm with a Q table that is updated using the bellman equation. This updating is done in order to maximize the rewards obtained. It is also dependent on various hyper parameters that are used to control the behavior of the algorithm. This maximum reward is then used to guide the agent to choose the best possible action at any given point of time. As the agent always chooses the best possible action based on the maximum reward, dynamicity is introduced in the game being played and also the winning probabilities are increased. This supports in treatment of patients as they would always win the game.

Keywords: reinforcement, schizophrenia, bellman.

1 Introduction

Human health is nothing but physical, emotional and also social wellbeing. Mental health comprises of our emotional, psychological, and social well-being. It affects how we think, feel, and act. It helps in handling stress and making choices. Mental health is vital at each stage of life, from childhood and adolescence through adulthood. Schizophrenia might cause chronic and serious mental disorder that can change the behavior of a person even in terms of his feelings and thinking abilities. They no more feel any connections to the realities of life. Although schizophrenia is not a frequently occurring mental disorder, the symptoms can affect the normal lifecycle of an individual to a very great extent. Symptoms of schizophrenia typically begin between ages sixteen and thirty.

Schizophrenia symptoms can be broadly classified into three types such as positive, negative, cognitive. Positive symptoms include psychotic behaviors that are not usually visible in healthy individuals. People with positive symptoms cannot majorly differentiate imagination and reality. Some major issues individuals face are Hallucinations, Delusions, thought disorders (prone to unusual or dysfunctional ways of thinking), movement disorders (leads to agitated body movements). Negative symptoms disturb normal emotional state of

ICCAP 2021, December 07-08, Chennai, India

DOI 10.4108/eai.7-12-2021.2314479

balance. Symptoms include: “Flat affect” (reduced expression of emotions via facial features or voice tone), reduced feelings of delight in standards of living, difficulties in starting and sustaining activities, reduced speaking. Cognitive symptoms of schizophrenia are not very chronic, but sometimes they are more serious and patients experience loss of memory and also changes in their ways of thinking. Symptoms include: Poor executive functioning which is the ability to take in information and make decisions based on them, Trouble in concentrating and also loss of attention and focus on anything. Patients also experience problems with working memory, which is the ability to make use of the data or information obtained immediately into practice

Many factors lead to the development of schizophrenia in individuals. Scientists believe that large amount of genes that are totally different from each other could cause schizophrenia in people; however, a single gene by itself can never cause schizophrenia. Genetic information does not really help in predicting on which individual is likely to develop schizophrenia.

It is also observed that when the genes that are present in the human body interacts with the environment around, it leads to developing mental disorders such as schizophrenia Some experts in the field also believe that faults in brain development before birth could also cause major concerns like schizophrenia. The major changes that happen during puberty have also been found to trigger many of the psychotic symptoms in individuals due to various biological and hormonal variations.[2] Treatments include antipsychotic medications that are usually consumed everyday either in pill or liquid form. Some antipsychotics are injections that are given once or twice a month or certain psychosocial treatments which would prevent relapses in patients. But the disadvantage is the side effects caused due to medication which affect other health conditions. People with schizophrenia can be trained by playing a video game implemented using reinforcement techniques to control the part of the brain linked to verbal hallucinations that would also help patients who do not respond to medications.

2 Reinforcement Based Learning

Reinforcement learning algorithm is about taking suitable action to maximize reward in any particular situation. The recursive process starts with the agent taking a random action for a particular state. On the basis of action taken at a particular state, rewards are updated to the agent by the environment. By this way the agent keeps updating the rewards in order to obtain the maximum reward.

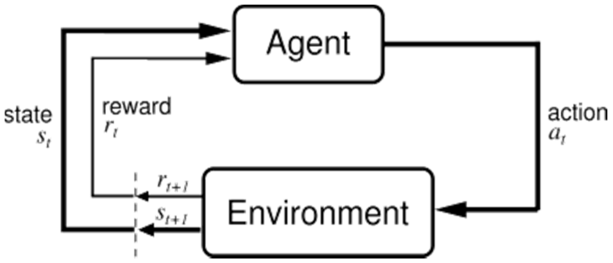


Fig 1.1 Reinforcement learning based recursive process

Terminologies

- **Agent:** Performs learning and decision-making processes.

- **Environment**: the place where agent learns and makes decision based on it.
- **Action**: It is performed by the agent on a particular state at a given time
- **State**: the state at which the agent belongs with respect to the environment
- **Reward**: A Scalar value which is a reward is provided for each action that the agent selects in the environment.
- **Policy**: it is used to map the actions to particular situations
- **Value function**: Long term reward is predicted with the value of the state where the state value is mapped to real numbers.
- **Function approximator**: Functions from training samples where used. Standard approximations include decision trees, neural networks, and nearest-neighbor methods
- **Markov decision process** (**MDP**): a probabilistic model based on a sequential decision problem, using which the states can be perceived exactly, and the current state and action selected determine a probability distribution on future states. The result of applying any action to any given state is not dependent on the previous actions but only on the current ones
- **Dynamic programming** (**DP**): provides solution methods to solve problems based on sequential decisions. This is solved with a cost structure that is composite
- **Monte Carlo methods**: it has a period of numerous approaches that helps in estimating the state value by running several trials starting at any particular state, and then averages the rewards in total that are obtained on these performed trials.
- **Temporal Difference** (**TD**) **algorithms**: methods to compare temporally successive predictions
- **Model**: Probability distributions are mapped as state-action pairs to states.

Types of reinforcement learning algorithm

Model free reinforcement learning: Policy optimization or policy-iteration strategies: In policy optimization methods the agent learns directly the policy function that helps in mapping the state to action.

Policies are divided broadly into two categories: deterministic and stochastic. Deterministic policy algorithms maps states to any of the particular actions taken. Stochastic policies give outputs in the form of probability distributions. This process is called as Partially Observable Markov Decision Process (POMDP)[1].

Q-learning or value-iteration methods: Q-learning is used to learn the action-value function Q(s, a):if the action is good enough to be taken at a particular state. Basically, a value which is of scalar type is associated with an action taken at a particular state. The following work flow represents the working of the algorithm.

Model based reinforcement learning: Model-based reinforcement learning contains a sturdy influence from control theory, and also the goal is to plan through an f(s, a) control function to decide on the optimal actions. The drawback of model-based strategies is that though they need a lot of assumptions and approximations on a given task, however it is restricted solely to these specific types of tasks.

3 Proposed System

Importing Gaming Environment

For the gaming environment, open gym AI tool is a bundle which is used to develop and compare RL algorithms . The structure of the agent is not considered. It is compatible with many of the numerical libraries such as Tensor Flow or Theano.[3] The gym library comprises of test problems and environments that help in performing reinforcement-based algorithms. It also provides a shared interface that helps in writing generalized algorithms. The game used for the implementation is frozen lake which requires that the agent never falls into any holes and reached the goal.

The step function based on the environment returns four values. These are:

- **Observation** (**object**): an object that is environment specific such as pixel data obtained from camera.
- **Reward** (**float**): It is like feedback which is obtained from the last performed action. The gage depends on the environments, but the final goal remains the same (to increase the total obtained reward)
- **Done** (**Boolean**): To predict if the environment requires resetting. Mainly not all but most tasks are divided into episodes. True results indicates that the episode is terminated.
- **Info** (**dict**): Debugging is used for diagnostic information.

Construction of Q table

Performing q learning algorithm involves the creation of a table or matrix in the form of [state, action]. It is initially initialized to zero. The updating of table is done after the completion of an episode. This table is then used as a guide in order to select for a given state the best possible action. This is done based on the q value

Setting up hyperparameters

Learning Rate: It can be defined as the amount by which the new value is accepted when compared to the old values. It involves taking the difference between the both and performing multiplication using the learning rate. The obtained result is summed to the q value previously obtained. This is then used to obtain the best possible action

Gamma: It is used to balance the immediate values and future reward values. Future rewards are based on this discounting factor. The value ranges from 0.8 to 0.9

Reward: After the operation has been performed on a given state, a updated score is received. The rewards obtained can occur at any time of the step.

Max: it is used to provide the maximum value of the future reward. This is then applied to the current state. The current action will be influenced by the possible future bonus. Allocation of future rewards to current actions taken helps the agent in identifying the action that would give the highest return

Exploration rate: Exploration is exploring more and more information about the environment. Exploitation is exploiting identified data to maximize the reward

Q Learning algorithm

Q learning algorithm helps in finding the best possible action that could be taken on any given state. It is an off-policy learning algorithm because the actions that are learnt are from outside the current policy. It takes random actions therefore there is no need of any particular or defined policies. The main focus is on maximising the future rewards obtained. After this process the agent interacts with its surroundings in order to make updates to the [state, action] pairs present in the Q table

Agent interaction: Agent's interaction with the environment takes place in two ways. In the first way the Q table is referred which provides a list of all possible actions that could be performed on any given state. After this the best action is selected by the agent based on the maximum value. This process is termed as exploitation. The term exploitation is used as it makes use of the existing information in order to make a decision. The second way involves taking actions randomly which is termed as exploring. Random selection of actions helps the agent to explore more and also take new states. The balance between exploration and exploitation is brought using the epsilon value. This epsilon value can be set by the users

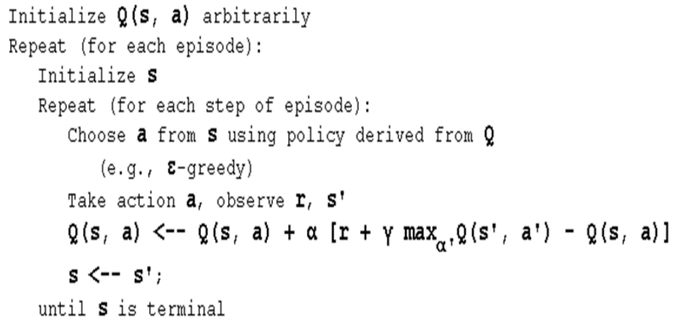
Updating Q table: Updating is done when an episode is done or reaches some terminal point. Agent learning does not take place in a great way after the end of an episode. But exploration and exploitation help the agent to learn and update the values into optimal ones.

Here are the 3 basic steps:

Agent starts in a state (s1) takes an action (a1) and receives a reward (r1)

Agent selects action by referencing Q-table with highest value (max) OR by random (epsilon, ε)

Update q-values



```
Initialize Q(s, a) arbitrarily
Repeat (for each episode):
   Initialize s
   Repeat (for each step of episode):
      Choose a from s using policy derived from Q
         (e.g., ε-greedy)
      Take action a, observe r, s'
      $Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{\alpha'} Q(s', a') - Q(s, a)]$
      s <-- s';
   until s is terminal
```

Game Play

The updated Q table consists of state action pairs with the maximum rewards. This Q table is then used as a reference for the agent to play. The finest possible act for any particular state is taken. This increases the winning probability of the agent. Dynamicity is also introduced in the game. The agent keeps adapting based on the action, reward and the environment.

4 Implementation Results

Importing Gaming environment: The frozen lake game environment which is available in open gym toolkit is imported and used

env = gym. Make("FrozenLake-v0")

Construction of Q table: When q-learning is performed a q-table or matrix that follows the shape of [state, action] is created and initialized to zero. The updating of table is done after the completion of an episode. Q table is then used as a guide to choose for a given state the best possible action. This is done based on the q value. Q table is nothing, but a fancy name given for a simple type of lookup table that calculates and displays the maximum value of future rewards for an action at any particular state. Bellman equation involving various hyper parameters is used to update the rewards in the Q table.

Setting the hyper parameters: Setting the right hyper parameters are important to get the maximum expected reward. Experiments were performed in order to identify the best range of hyper parameters for better performance.

Exploration rate: Exploration is exploring more and more information about the environment. Exploitation is exploiting identified data to maximize the reward. When exploration rate is too less then there is no initial random action taken. The action is taken based on the maximum reward and initially the entire Q table has values as zero. Therefore, a minimum exploration rate is required so that the agent can initially start with some random action and over the course start looking at the table for maximum rewards. Higher exploration rates also cause the system to fail as every time the agent starts taking random actions without referring the table for the maximum reward.

The comparison of the rewards obtained based on various exploration rates are given in table no 1.1

Table 1.1 Reinforcement learning based recursive process

Exploration Rate	Score
0.2	0
1.0	0.46
2.0	0.48
4.2	0.46
5.2	0.45
8.2	0.44

Exploration rate in the range (1.0 to 2.0) was found to give maximum reward.

Discount rate: It is used for balancing immediate values and potential values of reward. The element of discounting is extended to future rewards. The value ranges from 0.8 to 0.9. As the agent plays the game, more rewards are given to the closer actions taken or the initial levels. The goal is to make sure the agent wins the game with least number of steps and at the earlier level. Therefore, a discount rate is associated along with the steps taken far away from the initial source. Thus, the rewards decrease when the number of steps taken is greater. Also, very high discount factor wouldn't work properly as the agent at some point of time will reach the stages at the further levels

Table no 1.2- Comparing discount rates to obtain the rewards

Discount Rate	Score
0.95	0.46
0.86	0.37
1	0.0176

Discount rate in the range (0.95 to 0.99) was found to give maximum reward

Learning Rate: It can be defined as the amount by which the new value is accepted when compared to the old values. It involves taking the variance between the both and performing multiplication using the learning rate. The obtained value is then added to the q value previously obtained. This is then used to obtain the best possible action

Table no 1.3- Comparing learning rates to obtain the rewards

Learning Rate	Score
0.7	0
0.8	0.46
0.9	0.0

The best learning rate obtained is 0.8

Q Learning Algorithm: Q learning algorithm helps in finding the best possible action that could be taken on any given state. It is an off-policy learning algorithm because the actions that are learnt are from outside the current policy. It takes random actions therefore there is no need of any particular or defined policies. The main focus is on maximising the future rewards obtained. After this process the agent interacts with its surroundings in order to make updates to the [state, action] pairs present in the Q table. The updation is done using the bellman equation (1)

Bellman equation

Q[state, action] = Q[state, action] + lr * (reward + gamma * np.max(Q[new_state, :]) — Q[state, action])

------ (1)

Game Play: The updated Q table consists of state action pairs with the maximum rewards. Using the Q table, the best possible action for any particular state is taken. This increases the winning probability of the agent. Dynamicity is also introduced in the game. Based on the behavior, reward and environment, the agent continues to adapt. Experiments were carried out using the best chosen and worst chosen hyper parameters. The best-chosen hyper parameters gave good results with high scores and more winning probabilities and the worst chosen hyper parameters gave low rewards and the winning probabilities were decreased.

Table no 1.4 – Considering the worst chosen hyper parameters for the game play

Exploration Rate	0.2
Discount Rate	0.95
Learning Rate	0.8

All the episodes ended up in a hole and did not reach the goal

Table no 1.5 – Results of the game play using worst chosen hyper parameters

EPISODE 0 (Left)	EPISODE 1 (Left)	EPISODE 2 (Left)
SFFF	SFFF	SFFF
FHFH	FHFH	FHFH
FFFH	FFFH	FFFH
HFFG	HFFG	HFFG
Number of Steps 2	Number of Steps 17	Number of Steps 5

Table no 1.6 – Considering the worst chosen hyper parameters for the game play

All

Exploration Rate	2.0
Discount Rate	0.99
Learning Rate	0.8

Table no 1.7 – Results of the game play using best chosen hyper parameters the episodes resulted in a goal

EPISODE 0 (Down)	EPISODE 1 (Down)	EPISODE 2 (Down)
SFFF	SFFF	SFFF
FHFH	FHFH	FHFH
FFFH	FFFH	FFFH
HFFG	HFFG	HFFG
Number of Steps 75	Number of Steps 12	Number of Steps 20

Conclusion

The Q learning algorithm involves a Q table which is updated every time an action is performed on a particular state. It also depends on various hyper parameter values. Experiments were conducted on various values of each hyper parameters from which the best and worst range of hyper parameters were chosen. The best chosen hyperparameters gave better results with increased probabilities of winning.

References

[1] Torrado, R. R., Bontrager, P., Togelius, J., Liu, J., & Perez-Liebana, D. (2018, August). Deep reinforcement learning for general video game AI. In 2018 IEEE Conference on Computational Intelligence and Games (CIG) (pp. 1-8). IEEE.

[2] Jeste, D. V., & Maglione, J. E. (2013). Treating older adults with schizophrenia: challenges and opportunities. Schizophrenia bulletin, 39(5), 966968.

[3] Sodhi, S. (2017). AI for Classic Video Games using Reinforcement Learning.

[4] Li, Y. J., & Fan, F. Y. (2006, January). Classification of schizophrenia and depression by EEG with ANNs. In 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference (pp. 2679-2682). IEEE.

[5] Arbabshirani, M. R., Castro, E., & Calhoun, V. D. (2014, August). Accurate classification of schizophrenia patients based on novel resting-state fMRI features. In 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 6691-6694). IEEE.

[6] Barry, J. (2019). A Deep Learning Approach to Diagnosing Schizophrenia.

[7] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

[8] Shelton, J. O., & Arabnia, H. R. (2012, April). Brain imaging for diagnosis of schizophrenia: challenges, successes and a research road map. In 2012 Ninth International Conference on Information Technology-New Generations (pp. 578-583). IEEE.

[9] Iter, D., Yoon, J., & Jurafsky, D. (2018, June). Automatic detection of incoherent speech for diagnosing schizophrenia. In Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic (pp. 136-146).

[10] Allende-Cid, H., Zamora, J., Alfaro-Faccio, P., & Alonso-Sánchez, M. F. (2019). A Machine Learning Approach for the Automatic Classification of Schizophrenic Discourse. IEEE Access, 7, 45544-45553.

[11] Kalmady, S. V., Greiner, R., Agrawal, R., Shivakumar, V., Narayanaswamy, J. C., Brown, M. R., ... & Venkatasubramanian, G. (2019). Towards artificial intelligence in mental health by improving schizophrenia prediction with multiple brain parcellation ensemble-learning. NPJ schizophrenia, 5(1), 2.

[12] Qiao, C., Lin, D. D., Cao, S. L., & Wang, Y. P. (2015, November). The effective diagnosis of schizophrenia by using multi-layer RBMs deep networks. In 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 603-606). IEEE.

[13] D. S. Vijayan, A. Mohan, J. J. Daniel, V. Gokulnath, B. Saravanan, and P. D. Kumar, “Experimental Investigation on the Ecofriendly External Wrapping of Glass Fiber Reinforced Polymer in Concrete Columns,” vol. 2021, 2021.