

The background of the slide is a complex, abstract network diagram. It consists of numerous small black dots (nodes) connected by thin, light gray lines (edges). The connections are dense and form a web-like structure that fills the entire frame. Some nodes are more prominent than others, and the lines vary in thickness, creating a sense of depth and complexity.

# RL Poker

Negru Liviu-Bogdan

Judet Daniel Ilie

Octavian Doncea

Acest proiect  
urmărește construirea  
unui agent de  
reinforcement learning  
capabil să învețe  
plecând de la 0 jocul de  
poker sub regulile de  
Limit Texas Hold'em.





Regulile jocului sunt aceleasi ca la varianta obisnuita de Texas Hold'em, cu exceptia faptului ca suma de bani ce poate fi pariata la un anumit moment este predeterminata in functie de runda in care se afla jocul, si se calculeaza ca si incremente ale sumei big blind pariata initial

# Tehnologii folosite

- ◆ Pentru implementarea elementelor de machine learning ,au fost folosite librariile Pytorch si Tensorflow
- ◆ Pentru partea de environment si logica a jocului a fost folosita libraria RLCard



# Abordarea problemei

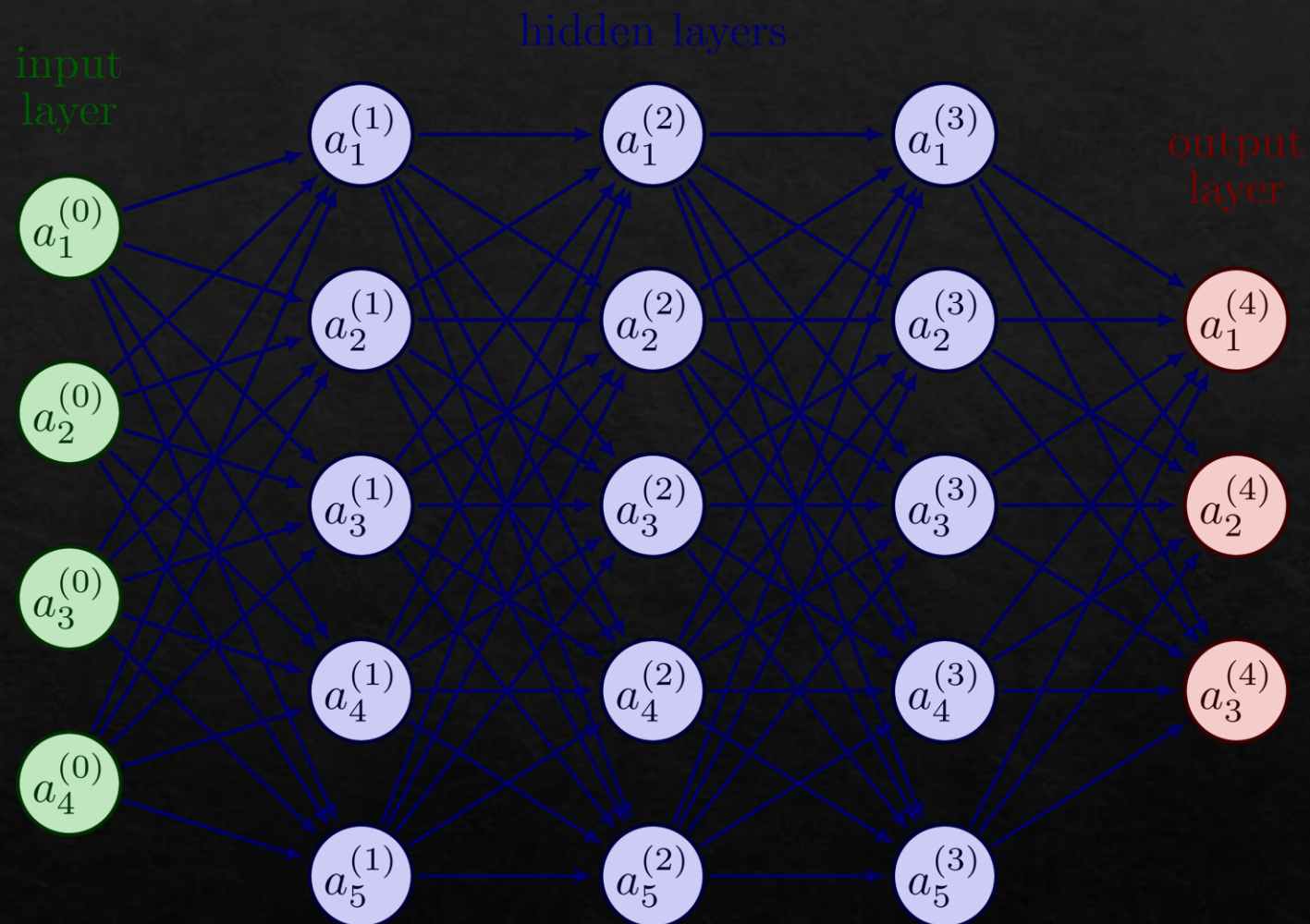
- ◇ Jocul de poker poate fi privit ca un proces decizional markov partial observabil , agentul nu putand observa niciodata starea completa a environment-ului.
- ◇ Asadar , este imposibila determinarea cu exactitate a unei politici optime pentru a parcurge mediul.
- ◇ In plus,desi finit,jocul de poker prezinta un numar imens de stari posibile,astfel abordarea clasica de Q learning este imposibila

- ◆ Solutia pe care o propunem este implementarea unui algoritm de DeepQ Learning., in care valoarea functiei  $Q$  in cazul fiecarei actiuni posibile este generata prin procesarea starii curente de catre o retea neurala.
- ◆ Agentul dispune de o 'memorie' proprie (o coada formata din tupluri ce contin starea , actiunea , reward , starea urmatoare si faptul daca episodul este terminat sau nu) din care ,de la un anumit moment,parti aleatoare sunt folosite pentru a antrena reteaua neurala din cadrul agentului nostru.



# Structura rețelei neurale

- ◆ Reteaua neurala folosita in cadrul acestui proiect este una de tip fully-connected avand doar un singur hidden layer , iar functia de activare folosita este de tipus sigmoid.



A fost aleasa functia sigmoid drept functie de activare datorita faptului ca ea este "abrupta " in jurul valorii 0 .Astfel, discrepanta intre valorile ce intra in retea va creste, ducand astfel si la o evidentiere mai mare in timp intre situatiile favorabile si nefavorabile.



# Logica de parcursere a jocului

- ◆ La fiecare runda environmentul jocului de poker se va reseta. Agentul va lua de fiecare data o actiune pe baza state-ului ce i se atribuie. Fiecare dintre actiuni va oferi un reward cu diverse valori, in functie de rezultatul obtinut. Daca runda s-a terminat valoarea sa o sa fie 0, in schimb daca agentul castiga v-a obtine o valoare pozitiva. Daca acesta o sa piarda valoarea o sa fie una negativa.

- ◆ La fiecare pas se memoreaza un set compus din starea actuala, actiune, reward-ul obtinut, actiunea urmatoare si daca s-a terminat sau nu partida; toate acestea prin intermediul variabilelor state, action, reward, next\_state, respectiv done.
- ◆ Odata ce o partida a fost terminata agentul va folosi tot ceea ce a memorat pentru a se antrena.

- ◆ Functia de loss aleasa este Mean Squared Error, deoarece aceasta are tendinta de a "pedepsi" mai tare agentul pentru erorile facute.



Rezultate obtinute/Concluzii  
(exprimare libera pe cod la  
prezentare)

# Bibliografie

- ◇ Using Artificial Neural Networks to Model Opponents in Texas Hold'em ,Aaron Davidson
- ◇ RLCard:A Toolkit for Reinforcement Learning in Card Games Daochen Zha , Kwei-Herng Lai , Yuanpu Cao , Songyi Huang , Ruzhe Wei , Junyu Guo, Xia Hu
- ◇ Libraria RLCard: <https://github.com/datamllab/rlcard>
- ◇ <https://rlcard.org/>
- ◇ Building Poker Agent Using Reinforcement Learning with Neural Networks Annija Rupeneite Faculty of Computing, University of Latvia
- ◇ Proiectul poate fi gasit aici: [GitHub](#)