САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Направление: 01.03.02 Прикладная математика и информатика

ООП: Прикладная математика, фундаментальная информатика и программирование

**Отчёт по проделанной работе**

**Тема задания:** решение задач методами машинного обучения в среде с неполной информацией на примере игры в покер

Выполнил: Хлопин Игорь Владимирович 19Б05

Руководитель научно- исследовательской работы: Гришкин Валерий Михайлович

Санкт-Петербург

2022

**Содержание:**

1. **Введение** ………………………………………………………………...………….............................3
2. **О стратегиях в играх**………………………………………………..………………………...….….3
3. **Об игре в покер**………………………………………………..…………………………….…….….4
4. **Исследуемая задача**………………………………………………..………………………….….….4
5. **Абстракция для неполной информации**………………………………………………..…...…….5
   1. Абстракция действия………………………………………………..……………………………5
   2. Абстракция информации………………………………………………..…………………….…5
6. **Методы обучения модели**………………………………………………..………………………….6
   1. Эмулирование игры противников………………………………………………..……...………6
   2. Игра модели с самой собой………………………………………………..……………..………6
7. **Представление данных для обучения и архитектура моделей**……………………………..….7
   1. Метод с эмулирование игры противников……………………………………………...………7
      1. Префлоп (первая стадия игры) …………………………………………………...………7
      2. Постфлоп (последние 3 стадии игры) ………………………………………...…………8
8. **Результаты обучения**………………………………………………..…………………………….…9
   1. Метод с эмулирование игры противников………………………………………………..….…9
9. **Вывод**………………………………………………..…………………………………………….….11

**Ссылки**………………………………………………………………………………......………….……12

**Введение**

В последние годы в области искусственного интеллекта ИИ были достигнуты большие успехи и игры часто служат сложными задачами ориентирами и вехами для прогресса. Покер десятилетиями служил такой сложной проблемой. Большинство прошлых успехов в таких играх включая покер, были ограничены вариантом для двух игроков. Однако, в покер в частности, традиционно играют более двух игроков. Многопользовательские игры представляют собой фундаментальные дополнительные проблемы, помимо тех, которые существуют в играх для двух игроков, а многопользовательский покер является признанной вехой в области искусственного интеллекта.

Покер на протяжении десятилетий служил сложной задачей для областей искусственного интеллекта ИИ и теории игр. Хотя покер был полезен в качестве эталона для новых методов искусственного интеллекта и теории игр, проблема скрытой информации в стратегических настройках не ограничивается играми. Концепции равновесия фон Неймана и Нэша были применены ко многим реальным проблемам таким как аукционы, кибербезопасность и ценообразование.

Последние десятилетия способности систем искусственного интеллекта играть в игры всё больше прогрессировали [2, 9, 10]. Однако в покере большинство предыдущих прорывов были ограничены настройками, в которых участвовали только два игрока [4]. В этой статье будет описана модель способная, побеждать игроков в безлимитном техасском холдеме для шести игроков, самом распространенном формате покера в мире.

**О стратегиях в играх**

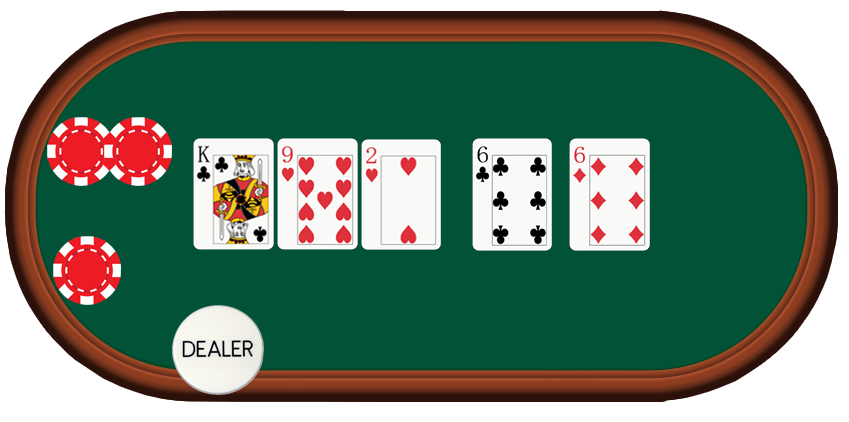
Искусственный интеллект достиг сверхчеловеческих результатов в таких играх как шашки, шахматы или го. Во всех этих играх участвуют только два игрока и они являются играми с нулевой суммой, это означает, что независимо от того, что выигрывает один игрок, другой игрок проигрывает. Каждая из систем искусственного интеллекта была создана путем попытки приблизиться к стратегии равновесия нэша. Равновесие по нэшу - это список стратегий по одной для каждого игрока в котором ни один игрок не может увеличить награду отклонившись к другой стратегии.

Так как покер является игрой с нулевой суммой, в данном проекте модель в процессе обучения будет так же стремиться к равновесию нэша. Но кроме этого, будут исследоваться методы построения стратегий на основании особенностей игры оппонентов.

Покер, в отличие от шахмат или шашек, как правило, является многопользовательской игрой, то есть в игре принимают участие множество игроков. В мультиагентной среде агенты взаимодействуют между собой и не всегда знают стратегии, цели и возможности своих оппонентов. Оптимальное поведение агента, максимизирующего свой результат в подобной среде, зависит от действий своих противников. Для построения эффективного агента в такой среде необходимо адаптироваться к действиям других игроков, моделируя их стратегии и обучаясь на основе их поведения.

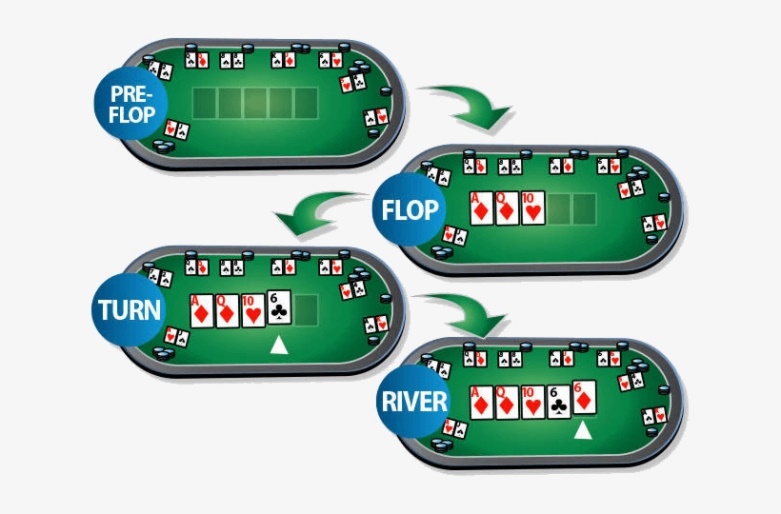
**Об игре в покер**

Техасский холдем - самая популярная разновидность покера. Принудительными ставками являются большой блайнд (BB) и малый блайнд (MB), MB = BB \* 0.5, их размеры должны быть установлены до начала игры в покер. После того как блайнды поставлены, игрокам выдается две частной карты в закрытом виде.



Существуют 4 раунда торговли:

* Префлоп - после того, как блайнды поставлены, игрок, сидящий слева от игрока, заплатившего большой блайнд, начинает торговлю. После окончания торговли дополнительно открывается три общие карты.
* Флоп - торговлю начинает игрок, поставивший малый блайнд. После окончания торговли дополнительно открывается одна общая карта в открытом виде.
* Терн - торговлю начинает игрок, поставивший малый блайнд. После окончания торговли дополнительно открывается одна общая карта.
* Ривер - торговлю начинает игрок, поставивший малый блайнд. После окончания торговли игроки открывают свои карты. Побеждает игрок, чья комбинация сильнее.



**Исследуемая задача**

**Цель исследования:** изучить и использовать методы машинного обучения в среде с неполной информацией для разработки модели, способной выбирать наиболее выгодные решения. Примером среде с неполной информацией является игра с нулевой суммой – покер.

**Проблемы:** главная проблема, ложащаяся на плечи искусственного интеллекта – это обработка огромного количества данных и игровых ситуаций.

**Используемые методы:** среда разработки – Colaboratory и Jupyter Notebook, язык программирования – Python, для построения модели использовались нейронные сети, состоящие из полносвязных слоёв, сети строились с использованием библиотеки Keras.

**Возможные применения:** алгоритмы, используемые для игры в покер, универсальны и в основном направлены на обучение агентов в средах с неполной информацией, поэтому эти алгоритмы можно перенести на множество приложений, где требуется принятие решений в среде с неполной информацией: как например в безопасности так и в маркетинге.

**Абстракция для неполной информации [1]**

В безлимитном техасском холдеме слишком много моментов для принятия решений, чтобы рассуждать о них по отдельности. Чтобы уменьшить сложность игры исключаются некоторые действия из рассмотрения, а также объединяются похожие точки принятия решений в процессе, это и называется абстракцией в покере. После абстрагирования распределенные точки принятия решения рассматриваются как идентичные.

В проекте рассматриваются два вида абстракции: абстракция действия и абстракция информации.

1. Абстракция действия

Абстракция действий уменьшает количество различных действий которые ИИ должен учитывать. В безлимитном техасском холдеме можно сделать любую ставку от 1 BB (большого блайнда) до всех фишек у игрока. Чтобы уменьшить сложность формирования стратегии модель рассматривает только, несколько различных вариантов торговли.

Три варианта торговли на любой стадии:

* Сброс карт (Fold)
* Чек/Колл (check/call; check – продолжаем играть без повышения ставки, при этом никто не делал ставку до; call – уравниваем ставку сделанную, да нашего хода)
* Бет/Рейз (Bet/Raise; bet – повышение ставки, при этом никто не делал ставку до; raise – переповышение ставки, сделанной до)

1. Абстракция информации

Другая форма абстракции, которая используется в проекте - это абстракция информации. При такой форме абстракции схожие точки принятия решения объединяются и обрабатываются одинаково. Эти объединённые наборы информации собираются вместе обрабатываются моделью одинаково, тем самым уменьшая количество различных игровых ситуаций. Для обобщения информации вводятся некоторые параметры, характеризующие её. Ниже приведён их список:

* Сила руки (Hand strength/HS) – используется на последних 3 стадиях торговли (флоп, тёрн, ривер), значение HS лежит в диапазоне [0, 1] и отражает отношение между количеством частных карт другого игрока, которые слабее наших, на общее количество всевозможных частных карт другого игрока.
* Диаграмма силы руки [3] – диаграмма отражает с какой вероятностью на выбранной стадии выбранная рука будет иметь определённую силу.

Пример некоторых параметров характеризующие стратегию игрока:

* VPIP – диапазон значений [0, 1], используется на первой стадии игры, описывает отношение между количеством раздач, в которых игрок внёс фишки в игру, и общим количеством раздач.
* PFR – диапазон значений [0, 1], используется на первой стадии игры, описывает отношение между количеством раздач, в которых игрок повысил ставку, и общим количеством раздач.
* AF – диапазон значений [0, 1], используется на последних трёх стадиях игры, описывает отношение между количеством раздач, в которых игрок повысил ставку (bet или raise), и общим количеством раздач.

**Методы обучения модели**

Большой проблемой в играх с неполной информацией, таких как покер, является нехватка баз данных для обучения. Поэтому, необходимо сыграть огромное множество игр, чтобы обучить модель. Рассмотрим решения данной проблемы:

1. Эмулирование игры противников

Для имитации имитация игры оппонента используются 8 базовых стратегий поведения игрока, которые можно описать параметрами, некоторые них были показаны выше. Эмуляция игроков позволяет проверить его реакцию на любое совершённое моделью действие.

Таким образом можно построить полное дерево всевозможных действий, которые модель принимает в раздаче. На каждой раздаче, обходя всё игровое дерево, можно выбрать наиболее выгодное действие, используя минимизацию сожалений. Такое действие можно обозначить вектором из трёх значений, где первый индекс обозначает сброс карт, второй – чек/колл, третий – бет/рейз, например уравнивание ставки (колл) будет обозначать вектор [0, 1, 0].

1. Игра модели с самой собой [5]

Данная стратегия так же рассчитана на минимизацию сожалений. Используется алгоритм CFR [6, 7, 8] – метод минимизации сожалений, это итеративный алгоритм игры модели с самой собой в котором она начинает с того, что играет полностью наугад, но постепенно совершенствуется, учась побеждать более ранние версии самой себя.

После каждой раздачи ИИ анализирует каждое принятое им решение и исследует насколько лучше, или хуже он бы поступил выбрав, вместо этого другие доступные действия. Такой обход всего дерева решения отличается от предыдущего метода только тем, что в векторе действия на месте наиболее выгодного хода стоит не единица, а число показывающее насколько данное действие выгоднее остальных [11], например вектор [30, 0, 0] говорит о том, что сбросить карты выгоднее на 30 BB (больших блайндов). Такое число так же называется сожалением, оно показывает насколько сильно игрок сожалеет о том, что не выбрал это действие в предыдущих итерациях. Выбирая наиболее выгодные действия модель минимизирует сожаления, поэтому данный метод и называется методом минимизации сожалений.

После того как было сыграно определённое количество раздач, модель, на основании полученных векторов с наиболее выгодными действиями, обновляет свои веса и вышеописанный процесс повторяется.

**Представление данных для обучения и архитектура моделей.**

1. Метод с эмулирование игры противников.
   1. Префлоп (первая стадия)

Выбор действия на первой стадии (префлоп) задан алгоритмически. Выбор основывается на диапазонах возможных рук, с которыми оппонент может совершить, то или иное действие.



На основании диапазонов вычисляется вероятность выигрыша модели против конкретно каждого игрока, а после объединяется в общую вероятность выигрыша в данной раздаче.

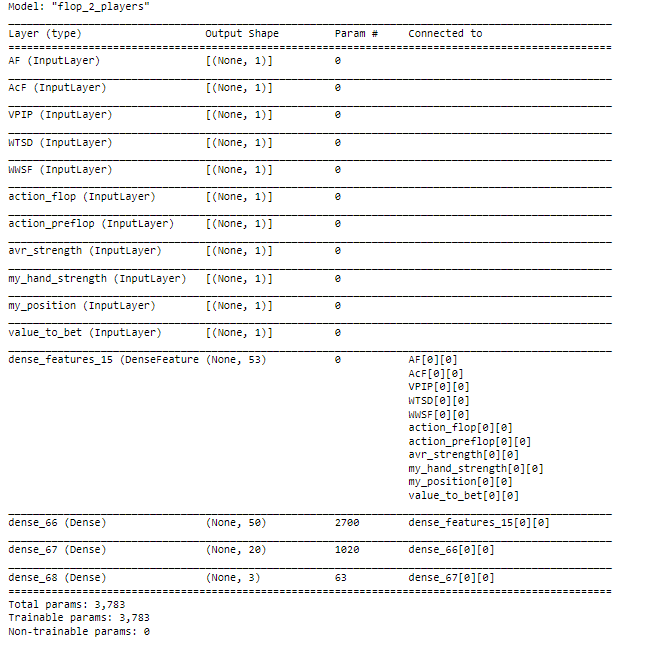
Зная вероятность победы можно вычислить математическое ожидание от той или иной ставки. Если мат. ожидание выше определённого значения, то делается колл или рейз, если же ниже, то карты сбрасываются (фолд).

* 1. Постфлоп (последние 3 стадии)

Для более точного обучения модели данные делятся на стадии и на количество игроков. На флопе, тёрне и ривере, за выбор действия отвечает модель, представляющая собой нейронную сеть, а точнее сказать модели. Для каждой стадии и для каждого количества игроков на данной стадии используется отдельная модель. Соответственно и наборы данных для каждой модели различны. Так как рассматриваем 3 стадии: флоп, тёрн и ривер, и 5 вариаций количества игроков, от 2 до 6 активных игроков, то за выбор действия в течение всей игры отвечают 15 моделей.

Так как рассматриваем 3 стадии: флоп, тёрн и ривер, и 5 вариаций количества игроков, от 2 до 6 активных игроков, то за выбор действия в течение всей игры отвечают 15 моделей.

Ниже приведен пример архитектуры сети используемой на второй стадии игры (флоп) при двух активных игроках за столом (модель и один оппонет):



От AF до value\_to\_bet показаны входные данные (параметры). Таким образом задача модели – это по описанным на картинке 13 параметрам выдать вектор размерности 3, значения которого обозначают вероятности действий в игровой ситуации, которая описывается данными параметрами, (1 индекс – фолд, 2 индекс – чек/колл, 3 индекс – бет/рейз).

Работу модели можно представить как: F(x) = y, x – набор из 13 параметров, y – результирующий вектор действия.

Набор, каждый элемент которого и есть описанные выше 13 параметров, разбит на отдельные векторы для того, чтобы их можно было подать на вход следующего слоя – tensor flow features. Далее следуют два полносвязных слоя по 50 и 20 нейронов соответственно и последний выходной слой из 3 нейронов.

Для всех нейронных сетей, используемых для реализации данного метода, выбраны одни и те же функции и метрики:

1. Функция потерь – категориальная кросс энтропия
2. Функция активации – у полносвязных слоёв это ReLu, у последнего слоя это Softmax
3. Оптимизатор – Adam, шаг обучения = 0.001
4. Метрика – accuracy

**Результаты обучения.**

1. Метод с эмулирование игры противников.

Далее будет представлено количество сыгранных раздач (обучающий набор) по стадиям игры и количеству активных игроков на данной стадии:

Флоп:

* 1. 10561 раздача
  2. 2293 раздачи
  3. 293 раздачи
  4. 18 раздач
  5. 0 раздач
* Тёрн
  1. 16047 раздач
  2. 2065 раздач
  3. 183 раздачи
  4. 11 раздач
  5. 0 раздач

Ривер:

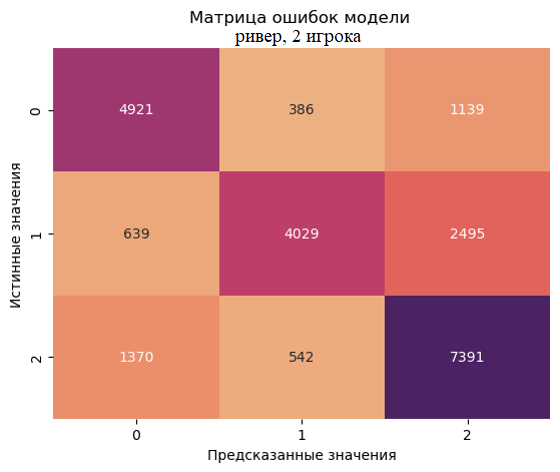
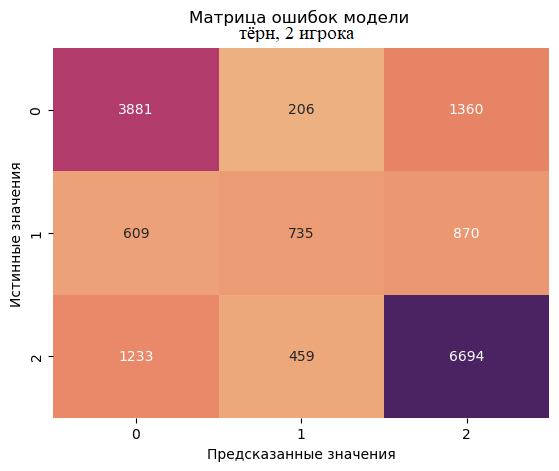
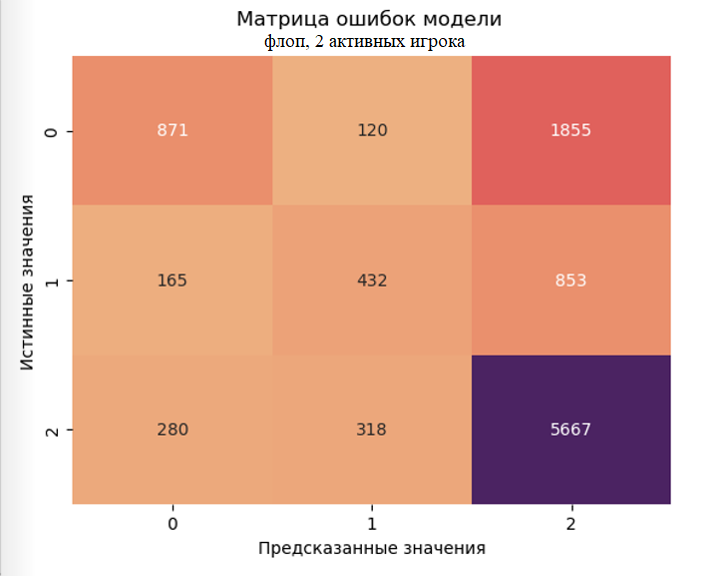
* 1. 22912 раздач
  2. 1782 раздачи
  3. 123 раздачи
  4. 18 раздач
  5. 0 раздач

Можно увидеть, что подавляющее большинство данных приходится на случай 2 активных игроков, поэтому ниже будут представлены результаты обучения моделей, используемых в случаях 2 активных игроков:

Значения функций потерь (cross-entropy) и метрик (accuracy):

* Флоп, 2 игрока: loss: 0.8733 - accuracy: 0.5919 - val\_loss: 0.6342 - val\_accuracy: 0.7609
* Тёрн, 2 игрока: loss: 0.7324 - accuracy: 0.6953 - val\_loss: 0.5009 - val\_accuracy: 0.7938
* Ривер, 2 игрока: loss: 0.6074 - accuracy: 0.7598 - val\_loss: 0.6210 - val\_accuracy: 0.7318

Матрицы ошибок:



Можно заметить, что подавляющая часть ошибок связана со случаями повышения моделью ставки, когда выгоднее сбросить карты, и наоборот - с ошибочным предсказанием сбросить карты, когда выгоднее поднять ставку.

**Вывод**

Метод с эмулированием игроков хоть и показал неплохие результаты в предсказывании наиболее выгодного действия в конкретной ситуации, использовать его в игре не представляется возможным. Обучением на игроках, с заранее прописанным поведением, сделало модель крайне негибкой, из-за этого модель часто путала ситуации, когда выгоднее сбросить карты или повысить ставку. Поэтому лучшим вариантом будет использование второго метода обучения модели, а именно – игра этой самой модели с собой.

Ссылки:

1. M. Johanson, N. Burch, R. Valenzano, M. Bowling, Evaluating State-Space Abstractions in Extensive-Form Games (2013)
2. N. Brown, T. Sandholm, Superhuman AI for multiplayer poker (2019)
3. S. Ganzfried, T. Sandholm, Potential-Aware Imperfect-Recall Abstraction with Earth Mover’s Distance in Imperfect-Information Games (2014)
4. N. Brown, S. Ganzfried, T. Sandholm, A Champion Two-Player No-Limit Texas Hold’em Poker-Playing Program (2015)
5. S. Srinivasan, M. Lanctot, V. Zambaldi, J. Pérolat, K. Tuyls, R. Munos, M. Bowling, Actor-Critic Policy Optimization in Partially Observable Multiagent Environments (2020)
6. M. Johanson, N. Bard, M. Lanctot, R. Gibson, M. Bowling, Efficient Nash Equilibrium Approximation through Monte Carlo Counterfactual Regret Minimization (2012)
7. M. Lanctot, K. Waugh, M. Zinkevich, M. Bowling, Monte Carlo Sampling for Regret Minimization in Extensive Games (2009)
8. Лаврик А.Е, Тропченко А.А, Cравнение методов агентного моделирования на примере игры в разновидность покера «техасский холдем» (2021)
9. E. G. Jackson, AAAI Workshop on Computer Poker and Imperfect Information (2016)
10. M. B. Johanson, Robust strategies and counter-strategies: from superhuman to optimal play, Ph.D. thesis, University of Alberta (2016).
11. A3C (Asynchronous Advantage Actor-Critic), url: <https://craftappmobile.com/a3c-asynchronous-advantage-actor-critic/#:~:text=A3C%20означает%20Asynchronous%20Advantage%20Actor,%2C%20отсюда%20и%20название%20–%20А3С> [Интернет-ресурс]