Санкт-Петербургский государственный университет

**Кафедра компьютерного моделирования и многопроцессорных систем**

**Хлопин Игорь Владимирович**

**Выпускная квалификационная работа бакалавра**

**Применение методов машинного обучения в среде с неполной информацией на пример игры в покер**

Направление 01.03.02

Прикладная математика фундаментальная информатика и программирование

Заведующий кафедрой,

Доктор технических наук,

Профессор Дягтярёв А.Б.

Научный руководитель,

Кандидат технических наук,

Доцент Гришкин В.М.

Рецензент,   
Кандидат физико-математических наук,   
Доцент Степенко Н.А.

Санкт-Петербург

2023

Оглавление

[**1.** **Введение** 3](#_Toc135576602)

[**2.** **Постановка задачи** 4](#_Toc135576603)

[**3.** **Обзор существующих решений** 4](#_Toc135576604)

[**4.** **Об игре в покер** 5](#_Toc135576605)

[**5.** **Абстракция для неполной информации** 7](#_Toc135576606)

[**6.** **Методы обучения моделей** 8](#_Toc135576607)

[6.1. Эмуляция человеческой игры 8](#_Toc135576608)

[*6.1.1.* *О методе* 8](#_Toc135576609)

[*6.1.2.* *Архитектура алгоритма эмитирующего игру человека* 9](#_Toc135576610)

[*6.1.3.* *Архитектура модели* 11](#_Toc135576611)

[6.2. Модель актёр-критик и метод самообучения 12](#_Toc135576612)

[*6.2.1.* *О методе и модели* 12](#_Toc135576613)

[*6.2.2.* *Архитектура модели актёр-критик* 12](#_Toc135576614)

[*6.2.3.* *Награды для обучения модели и функция потерь* 13](#_Toc135576615)

[*6.2.4.* *Алгоритм обновления весов модели в игре с предыдущими версиями себя* 14](#_Toc135576616)

[**7.** **Результаты обучения** 14](#_Toc135576617)

[*7.1.* *Результаты игры против базовых стратегий.* 15](#_Toc135576618)

[*7.2.* *Результаты игр моделей между собой.* 17](#_Toc135576619)

[**8.** **Заключение** 19](#_Toc135576620)

[Список литературы: 21](#_Toc135576621)

1. **Введение**

В последние годы в области искусственного интеллекта ИИ были достигнуты большие успехи и игры часто служат сложными задачами ориентирами и вехами для прогресса[7]. Покер десятилетиями служил такой сложной проблемой. Искусственный интеллект значительно продвинулся в решении задач в несовершенно информационных играх, включая покер.

Множество покерных алгоритмов используют комбинацию методов машинного обучения, игровой теории и алгоритмов поиска для принятия оптимальных решений в контексте несовершенной информации.

Такие достижения в искусственном интеллекте в области покера имеют значительный потенциал для применения в других областях, где несовершенная информация играет ключевую роль. Например, в бизнесе и экономике, принятие решений может также зависеть от неопределенных факторов, и как правило, не все факты доступны на момент решения. Таким образом, исследования в области искусственного интеллекта и покера могут иметь большое значение в различных секторах экономики, науки, технологий, и инновационных начинаниях

В последнее время игры в лимитный и безлимитный покер с участием двух игроков достигли экспертного уровня. Например, [3,8] решает хендз-ап безлимитный техасский холдем. В этих работах используется минимизация контрфактических сожалений для поиска по игровому дереву, но это требует огромных вычислительных ресурсов для решения в реальном времени. Игра в покер определяется как игра с нулевой суммой. Теоретически оптимальная стратегия игры с нулевой суммой для двух игроков это стратегия равновесия по Нэшу. Следовательно, решением для игры в покер двух игроков является аппроксимация стратегии равновесия по Нэшу.

Однако, когда в игру вступают большее количество игроков, эта задача становится гораздо более сложной. В играх с более чем двумя игроками не существует уникального равновесия по Нэшу, и задача нахождения оптимальной стратегии становится более сложной[10]. Даже если каждый игрок найдёт оптимальную стратегию, список их стратегий может и не быть оптимальным, как например в игре в киоск с лимонадом[11].

В целях улучшения эффективности обучения в таких сложных средах, исследователи разрабатывают методы, такие как глубокое обучение (Deep learning)[14] и обучение с подкреплением (Reinforcement learning или RL)[15].

1. **Постановка задачи**

Основной целью данной работы является исследование эффективности применения алгоритмов машинного обучения в среде с неполной информацией на примере покера. А также, методом сравнения алгоритмов между собой будет выявлен лучшая алгоритм для решения задач в средах с неполной информацией.

В данной работе будет рассмотрен метод многоагентного обучения с подкреплением[15]. Обучение с подкреплением обычно состоит из методов, основанных на ценности, и методов, основанных на политике.

Архитектура актёр-критик[4,9] объединяет в себе методы: основанный на политике (актёр) и основанный на ценности (критик). В архитектуре актёр-критик актёр принимает решения на основе определенной политики, которая меняется в процессе обучения. Актёр сначала получает состояние среды, а затем выбирает действие, которое должно быть выполнено на основе текущего состояния. Критик оценивает решения актёра, используя ценностную функцию, и предоставляет обратную связь, которую актёр использует для обновления политики.

Эта архитектура является гибкой и может быть адаптирована к различным задачам обучения с подкреплением. Именно поэтому в данной работе для обучения модели игре в покер использована архитектура актёр-критик.

Также, для сравнения эффективности вышеописанной архитектуры используется модель, обученная на игре против эмулирующих реальных игроков алгоритмов.

1. **Обзор существующих решений**

Одни из самых популярных решений в области многопользовательского покера это:

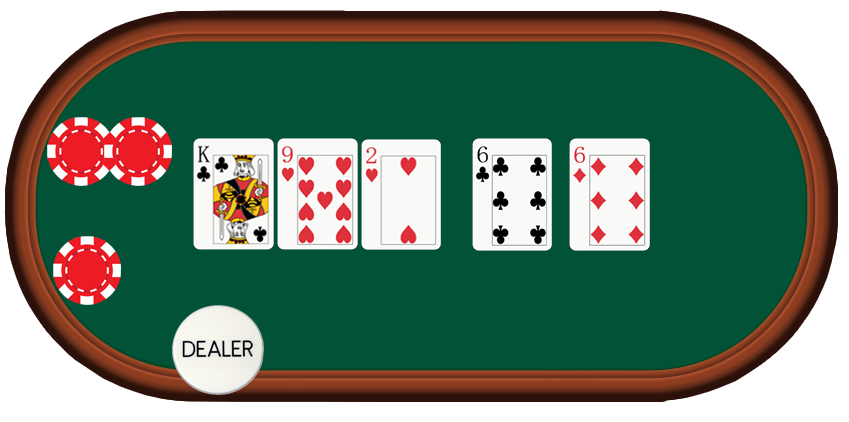
1. DeepStack[12] - это алгоритм для игры в покер и основан на глубоком обучении (deep learning). Модель была обучена на огромном количестве игровых данных, а специально разработанная архитектура нейронной сети позволяет ей делать умные и логичные решения в реальном времени. DeepStack был успешно протестирован в игре против профессиональных игроков и показал высокую эффективность.
2. Libratus[8] - это еще один алгоритм для игры в покер, который был разработан на основе комбинации различных методов машинного обучения, включая алгоритмы оптимизации линейного программирования и алгоритма Монте-Карло. Он был обучен на огромном количестве данных (более 15 миллионов рук) и успешно справился в игре против четырех профессиональных игроков, показав высокую точность принятия решений.
3. Pluribus[13] – это подход к обучению модели игры в покер, который был разработан на базе комбинации различных методов машинного обучения, включая обучение с подкреплением (reinforcement learning) и самообучение. Pluribus обучался на огромном количестве данных через тренировочные матчи против самого себя.

Но есть значительный недостаток, который встречается у каждого из вышеописанных методов – это высокая вычислительная сложность, как во время обучения, так и при вычислении предсказания. Этот недостаток может ограничивать их применимость в реальном времени и в реальной среде.

Поэтому описанный в данной работе алгоритм берёт на себя задачу избавиться от подобной проблемы, чтобы обеспечить применимость для работы в реальном времени.

1. **Об игре в покер**

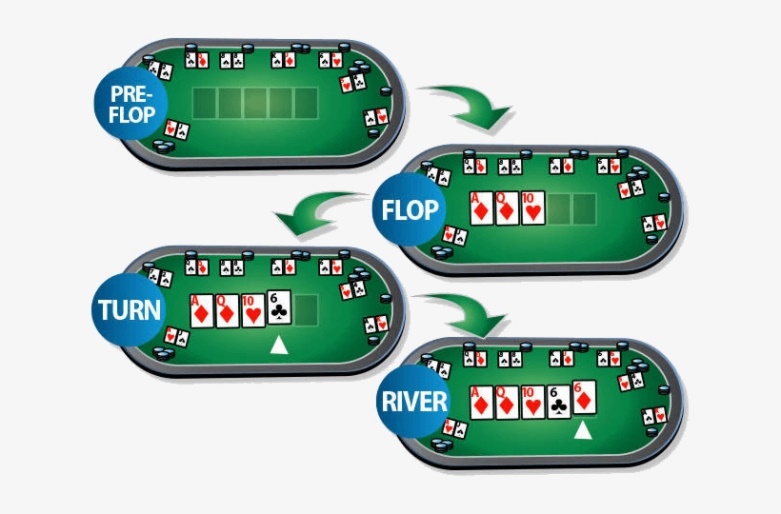
Техасский холдем - самая популярная разновидность покера. Принудительными ставками являются большой блайнд (BB) и малый блайнд (MB), MB = BB \* 0.5, их размеры должны быть установлены до начала игры в покер. После того как блайнды поставлены, игрокам выдаются по две карты в закрытом виде.



**Рис. 1.** Расположение блайндов за столом.

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.
* Тёрн - 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.



**Рис. 2.** Стадии торговли.

Торговля в покере представляет из себя 4 вида действий: фолд, колл, бет и рейз. Это основа стратегии игры в покер, которая позволяет игрокам контролировать размер банка и максимизировать свои выигрыши.

* Фолд – сдача карт и отказ от дальнейшего участия в игре.
* Колл (или чек) – уравнивание ставки, которую сделал соперник (или отказ от повышения ставки, если ставку так никто и не сделал).
* Бет – ставка, которую делает игрок, при условии, что до него никто не ставил.
* Рейз - дополнительная ставка поверх предыдущей ставки другого игрока.

Каждый игрок выбирает свою стратегию в зависимости от карт на руках и поведения других игроков. Цель - выиграть банк, сделав наибольшую ставку, заставив других игроков сделать фолд. Если к концу последней стадии (ривер) игроков осталось больше одного, то выигрывает тот, у кого комбинация личных карт и карт на столе сильнее.

1. **Абстракция для неполной информации**

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[1]. После абстрагирования распределенные точки принятия решения рассматриваются как идентичные.

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

1. Абстракция действия

Данная абстракция уменьшает количество рассматриваемых действий, которые ИИ должен учитывать. В безлимитном техасском холдеме можно сделать любую ставку от 1 BB (большого блайнда) до всех фишек имеющихся 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) – значение HS находится в диапазоне [0, 1] и отражает отношение между количеством частных карт другого игрока, которые слабее наших, и общего количества всевозможных частных карт другого игрока.
* Кластеры карт на доске – вместо того, чтобы учитывать всевозможные карты на доске, можно рассматривать только их группы.
* Диаграмма силы руки[2] – диаграмма отражает с какой вероятностью на выбранной стадии выбранная рука будет иметь определённую силу.

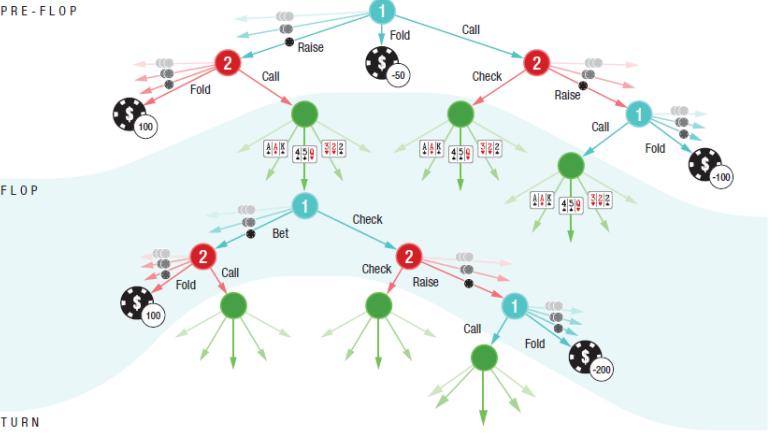
Пример некоторых параметров характеризующие стратегию игрока:

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

1. **Методы обучения моделей**
   1. Эмуляция человеческой игры
      1. *О методе*

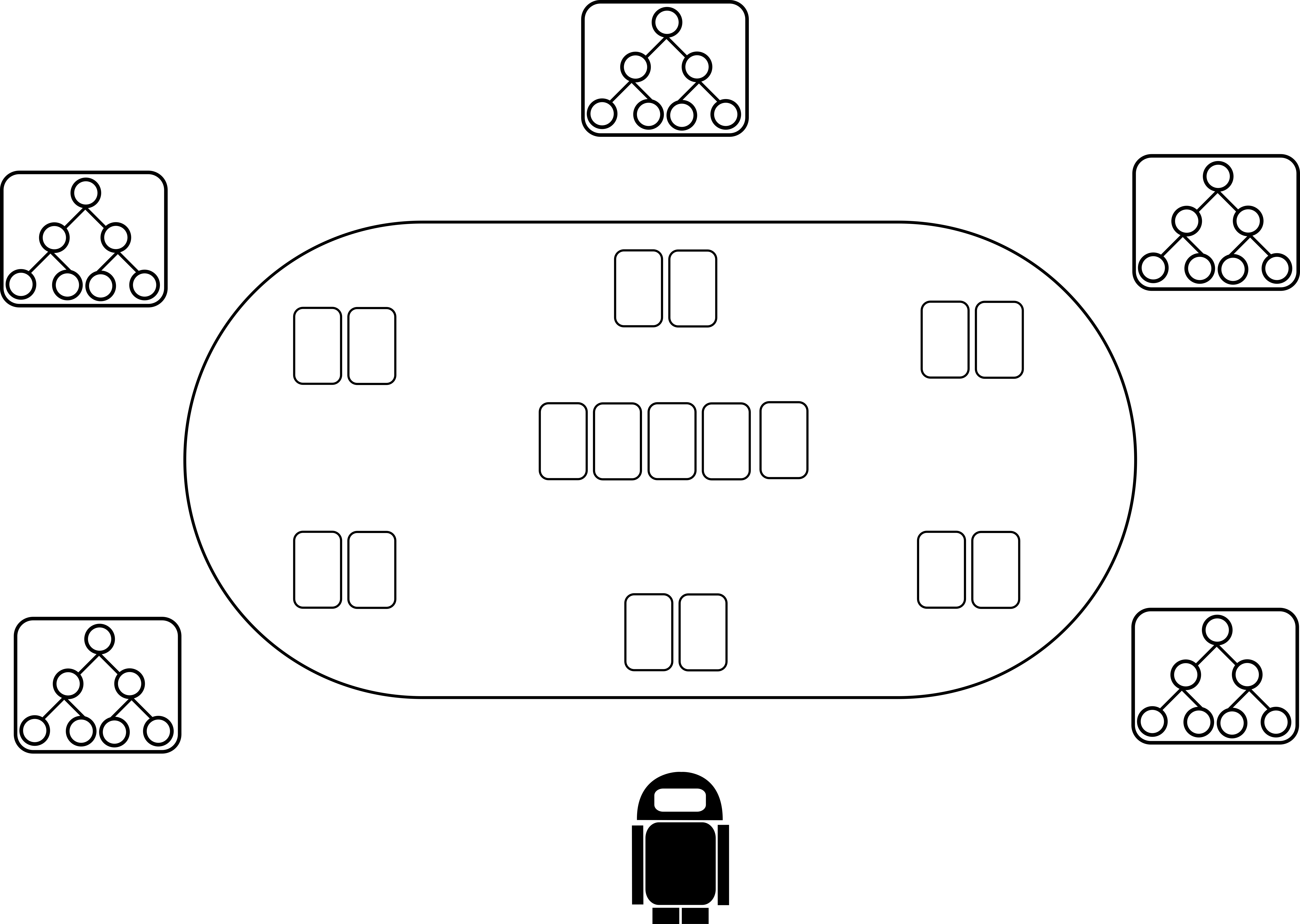
Данный метод позволяет решить проблему несовершенной информации в покере через эмуляцию человеческой игры противниками. То есть модель будет обучаться на игре против алгоритмов эмулирующий игру человека с определённой стратегией. Было отобрано 8 стратегий, которые использовались при эмуляции человеческой игры.

Тот факт, что все оппоненты лишь алгоритмы, эмулирующие игру человека, позволяет проверить реакцию противников на любое действие, совершённое моделью. Таким образом можно построить полное дерево всевозможных действий, которые модель может совершить в раздаче в раздаче. Обойдя игровое дерево, можно выбрать наиболее выгодное действие на каждой стадии, используя минимизацию сожалений. Пример такого дерева показан на рисунке ниже.



**Рис. 3.** Дерево всевозможных действий в раздаче.

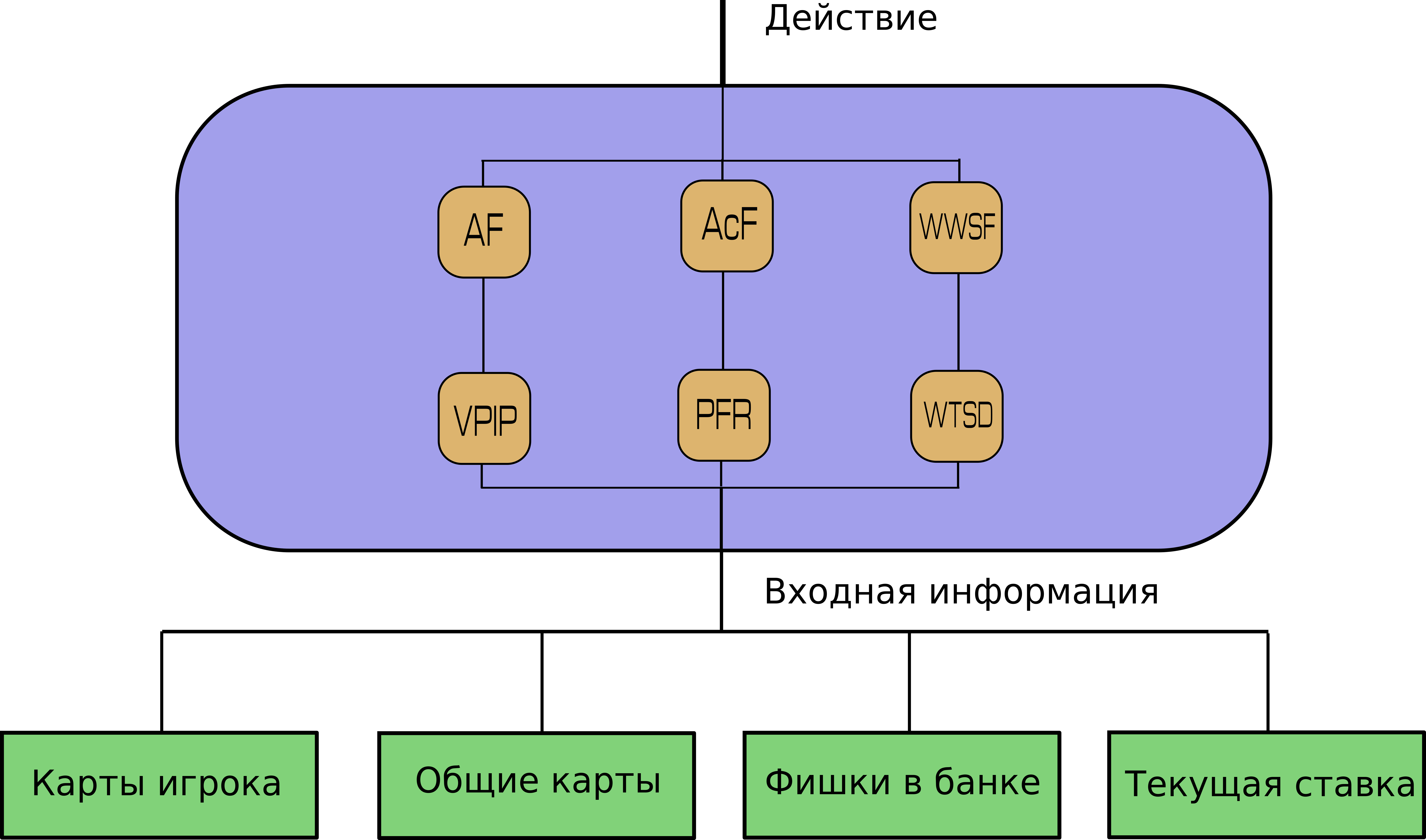
Процесс игры можно показать схемой:



**Рис. 4.** Схема стола с моделью и 5 алгоритмами.

На схеме обучаемая модель играет с 5 алгоритмами, эмулирующими игру человека с определённой стратегией.

* + 1. *Архитектура алгоритма эмитирующего игру человека*



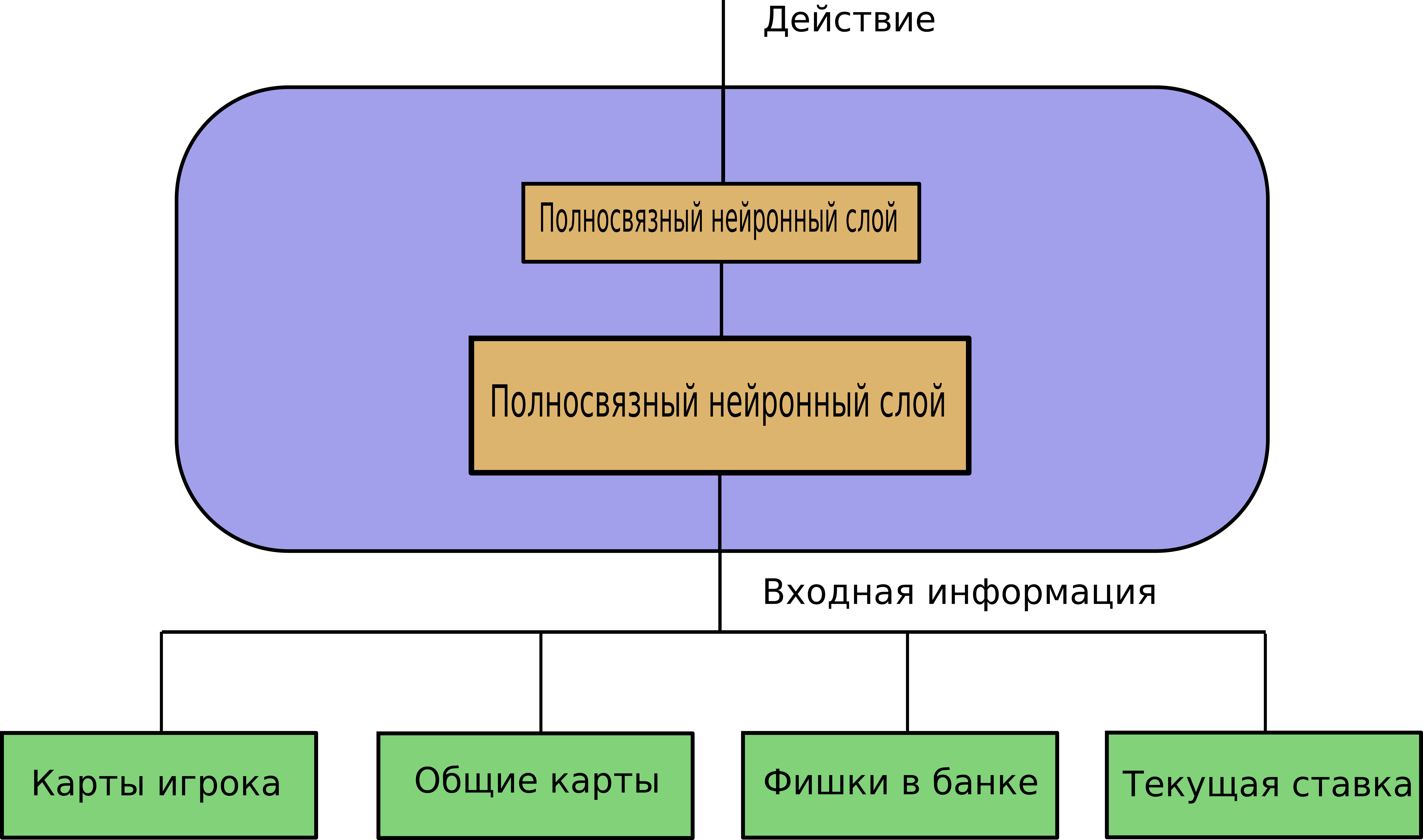
**Рис. 5.** Архитектура алгоритма эмитирующего игру человека.

Для имитации игры человека используется алгоритм, поведение которого зависит от значений его коэффициентов. Список используемых коэффициентов:

* VPIP – диапазон значений от 0 до 1, описывает отношение между количеством раздач, в которых игрок внёс фишки в игру, и общим количеством раздач.
* PFR – диапазон значений от 0 до 1, описывает отношение между количеством раздач, в которых игрок повысил ставку, и общим количеством раздач.
* WTSD – диапазон значений от 0 до 1, описывает отношение между количеством раздач, в которых игрок дошёл до вскрытия карт, и общим количеством раздач.
* AF – диапазон значений от 0 до 1, описывает отношение между количеством раздач, в которых игрок повысил ставку (bet или raise), и общим количеством раздач.
* AсF – диапазон значений от 0 до 1, описывает отношение между количеством раздач, в которых игрок повысил ставку (bet), и в которых сыграл чек.
* WWSF – диапазон значений от 0 до 1, описывает отношение между количеством раздач, в которых игрок выиграл раздачу, и общим количеством раздач.

Алгоритм выбирает одну из восьми самых популярных стратегий игры в покер с помощью варьирования вышеописанных коэффициентов. Стратегии различаются между собой степенью агрессивности разыгрывания карт, выбором карт для вхождения в игру, размерами ставок. Наличие у алгоритма возможности выбора стратегии розыгрыша своих карт позволяет модели обучиться игре против разных типов оппонентов.

* + 1. *Архитектура модели*



**Рис. 6.** Архитектура модели для обучения на алгоритмах, эмитирующих игру человека.

Теперь подробнее рассмотрим архитектуру модели, которая будет обучаться на игре против вышеописанных алгоритмов, имитирующих одну из восьми стратегий розыгрыша карт. Как показано на рисунке, модель, получая информацию о ситуации в игре, обозначим её вектором w, выдаёт результат работы нейронной сети, обозначим его a = F(w).

Результат работы нейронной сети представляет из себя нормализованный вектор размерности 3, где каждый элемент – это вероятность данного хода:

* Первый элемент – фолд
* Второй элемент – чек/колл
* Третий элемент – бет/рейз

Для обучения модели не хватает лишь метки к входным данным. Эта проблема решается с помощью дерева всех ходов (на рисунке выше), которые может совершить модель. Обойдя всё дерево можно составить вектор выигрышей для каждой игровой ситуации, обозначим его y. Таким образом для обучения модели имеются вектор вероятностей для каждого хода и вектор выигрышей для этих же ходов. Тогда функцию ошибки можно обозначить как L(a, y) = ln(y[i] / y[i] – a[i]), где i – индекс действия с наибольшим выигрышем.

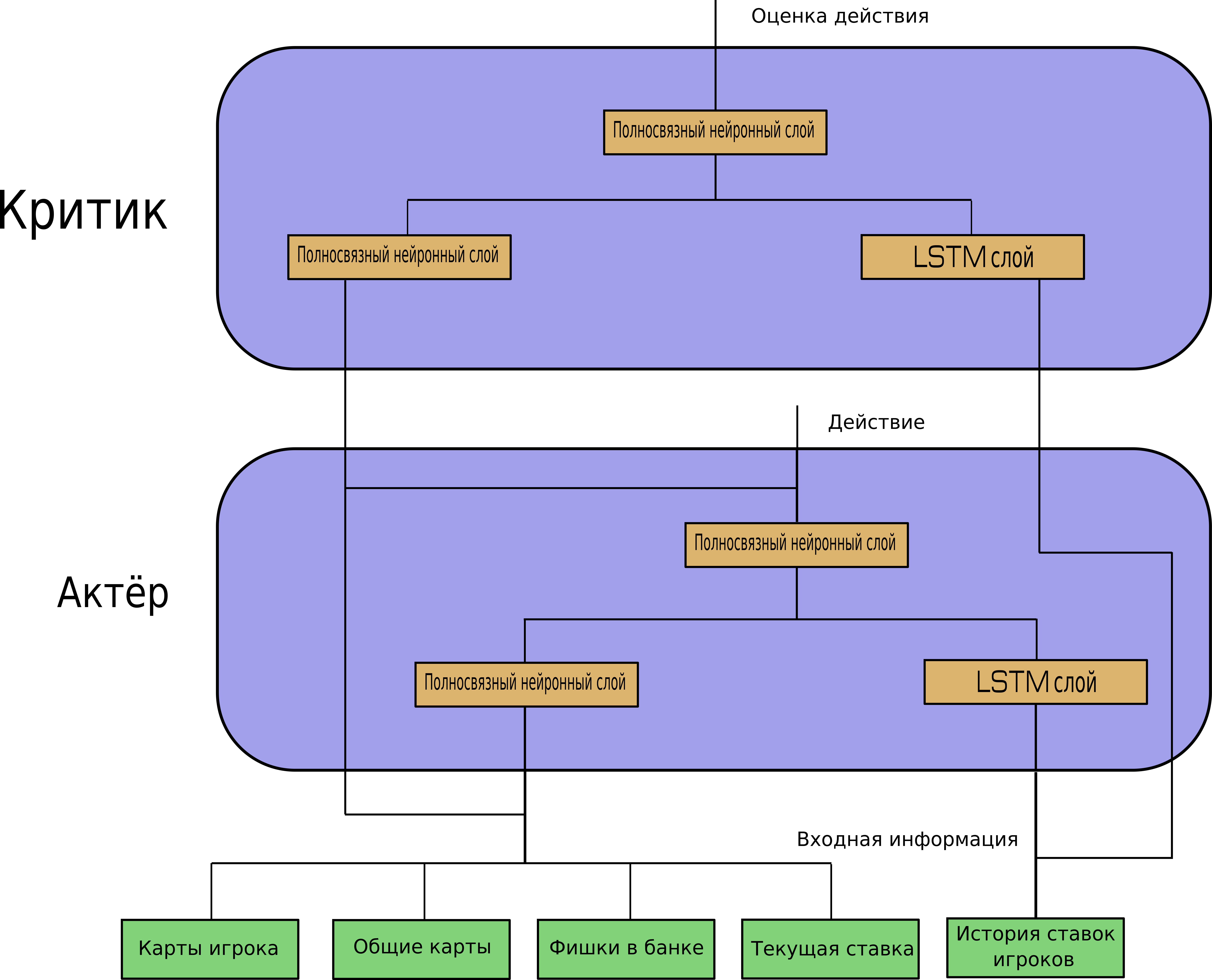
Например, пусть вектор выигрыша y = (-5, 4, -10), предсказание модели a = F(w) = (0.1, 0.3, 0.6), можно заметить, что элемент с индексом i = 2 (y[i] = 4) имеет наибольшее значение, значит этот ход принесёт наибольший выигрыш, второй элемент – это чек/колл, вероятность этого хода была a[i] = 0.3, следовательно функция ошибки будет равнна L(a, y) = ln(1 – 0.3).

* 1. Модель актёр-критик и метод самообучения
     1. *О методе и модели*

В этом разделе будет представлен метод обучения через игру модели с самой собой и будет представлена архитектура актёр-критик.

Обучение модели через игру с самой собой – ещё один способ решения проблемы нехватки информации для обучения. Кроме того, такой подход не ограничивает модель в обучении, как вышеописанный метод, так как модель постоянно пытается обыграть саму же себя, тем самым постоянно улучшается.

* + 1. *Архитектура модели актёр-критик*



**Рис. 7.** Архитектура модели актёр-критик.

На рисунке можно заметить, что модель состоит из двух сетей: актёра и критика, которые взаимодействуют друг с другом. Цель актёра – это рассчитать вероятности каждого из возможных действий (в данном случае 4), то есть результатом работы актёра будет нормализованный вектор a = F(w), где w – информация о текущей игровой ситуации (на картинке это входная информация).

В модели актёр-критик. Для сети критика основная задача состоит в том, чтобы оценить, разумно ли принимать выбранное актёром решение в данных условиях состояния игры. Для выполнения этой задачи критик запрашивает всю входную информацию о ситуации за покерным столом и результатом работы модели является оценка выбранного актёром действия, обозначим эту оценку как Q(w, a). Актёр же использует эту оценку, чтобы понять, какие действия ему следует выбирать в будущем. Он использует выходную информацию из сети критика для обновления своей стратегии и постоянно совершенствует свои действия, основываясь на полученных оценках. Таким образом, критик играет роль учителя, который дает актёру обратную связь о том, насколько хороши его действия в конкретной ситуации. Это помогает актёру постоянно улучшать свою игру и принимать более обоснованные решения.

* + 1. *Награды для обучения модели и функция потерь*

Прежде чем представить функцию потерь и принцип награждения модели, стоит уточнить, что процесс автономной игры в покер разбит между несколькими моделями. В покере 4 стадии игры, а активных игроков за столом может быть от 2 до 6, для каждой из стадий и для каждого возможного количества активных игроков используется отдельная модель. Таким образом для автономной игры используется 4 \* 5 = 20 моделей, каждая из моделей обучается на ситуациях, предназначенных именно для неё, например, модель, предназначенная для игры на флопе с 3 активными игроками (включая её саму), будет обучаться именно на таких ситуациях, в которых мы находимся на флопе с 3 активными игроками.

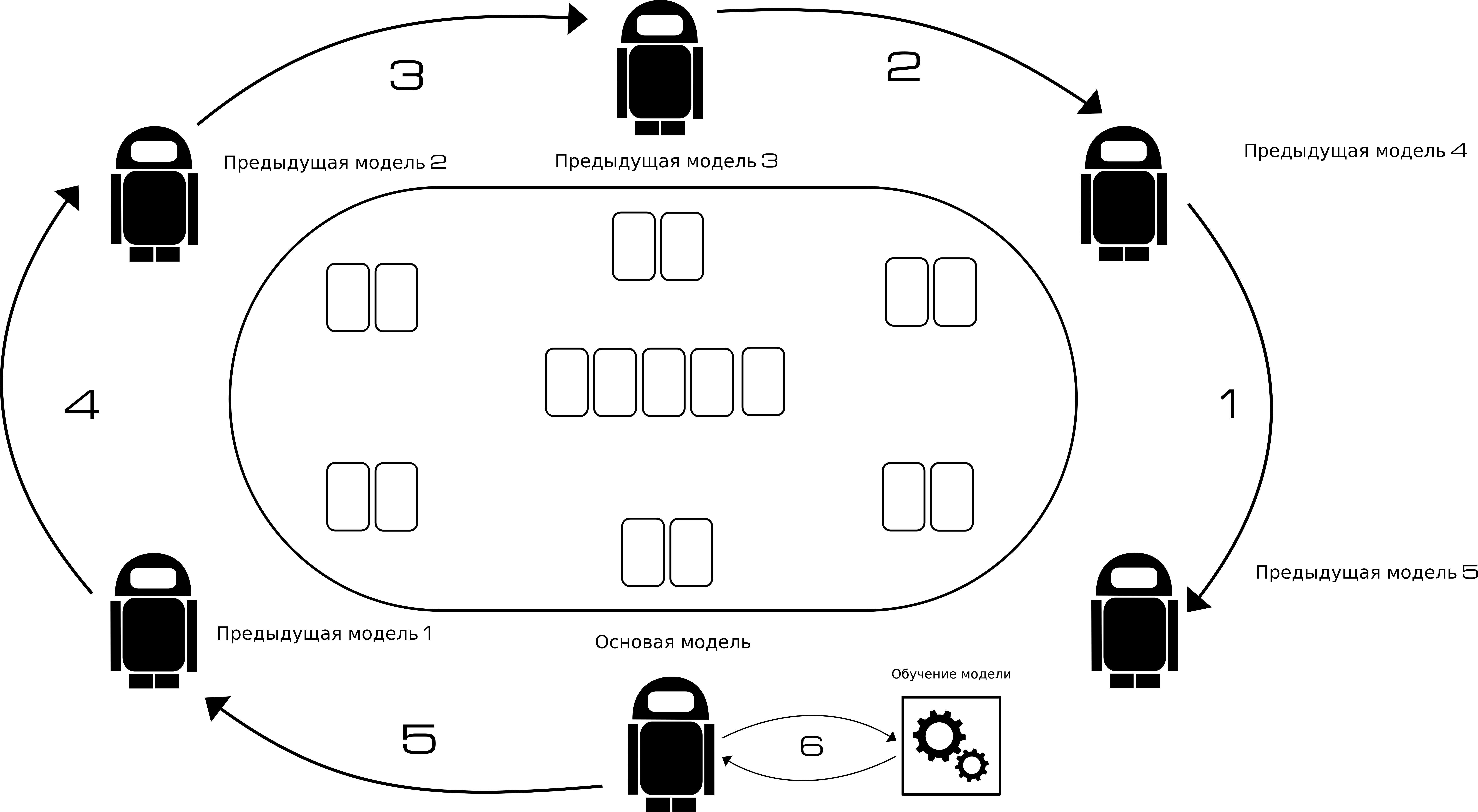
Для обучения с подкреплением модели актёр-критик необходимо ввести функцию награды. Награда представляет из себя число, показывающее насколько совершённое действие было выгодно или невыгодно для игрока. Пусть b1, b2, b3, b4 – ставки игрока на префлопе, флопе, тёрне, ривере соответственно, B = b\_1 + b\_2 + b\_3 + b\_4, M – выигрыш в данной раздаче (0 если игрок проиграл), тогда функция нагрыды примет вид:

r\_i = -b\_i + b\_i / B \* M

Полученная награда используется для обучения моделей актёр и критик. Функция потерь актёра выглядит как L(r\_i, a) = -ln(1 – a[j]) \* (r\_i – Q(w, a)), где Q(w, a) – результат работы критика, a[j] – вероятность выбранного действия. Функция потерь критика – это функция потерь Хьюбера для r\_i и Q(w,a).

* + 1. *Алгоритм обновления весов модели в игре с предыдущими версиями себя*

Процесс обновления модели можно увидеть на схеме, изображённой на картинке ниже.



**Рис. 8.** Схема алгоритма обновления моделей актёр-критик во время обучения.

Процесс можно описать так, по истечению заданного количества игр происходит смещение моделей по часовой стрелке, а на место основной модели встаёт новая модель, которая является результатом обучения основной модели на данных полученных в ходе заданного количества игр. Таким образом модель обучается обыгрывать предыдущие версии самой себя, что позволяет постоянно совершенствоваться и искать оптимальную стратегию игры приносящую наибольшие награды за предсказанные действия.

1. **Результаты обучения**

В этом разделе будут продемонстрированы результаты обучения моделей вышеописанными способами. Для оценки обученных моделей используется Техасский холдем с 6 игроками максимум. Чтобы явно противопоставить любые две политики, проверка результатов обучения подчиняется следующим правилам:

* Тестируемая политика действий против 5 тестирующих политик. При сравнении эффективности двух политик главный игрок использует политику, подлежащую оценке качества, и пять других игроков используют вторую политику в качестве тестирующей группы.
* Средний выигрыш после 1000 раздач. Для оценки результативности тестируемой политики используется метрика (ББ/К), где ББ – общий выигрыш в больших блайндах, К – количество раздач.

Такой подход позволяет объективно оценить качество обученной модели, так как случайный фактор, который в покере играет немаловажную роль, будет сведён к минимуму.

* 1. *Результаты игры против базовых стратегий.*

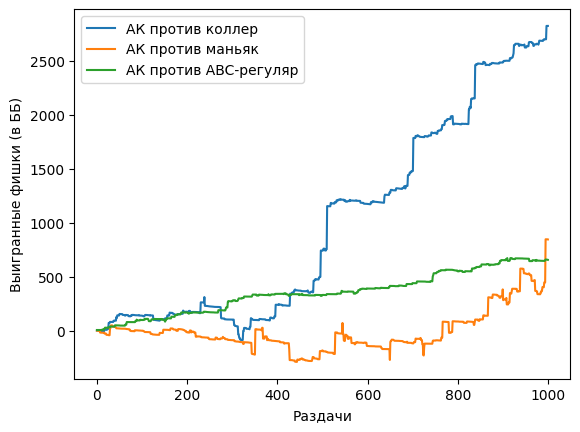
Теперь будут рассмотрены результаты игр модели Актёр-Критик (АК), обученную на игре с самой собой, и модели, обученной с использованием эмуляции человеческой игры (ЭЧИ). Играть модели будут против шаблонных стратегий поведения:

* Автоответчик. Стратегия, которая почти исключает повышения ставки вне зависимости от имеющейся руки. Такой игрок отвечает колл на почти все ставки оппонентов.
* Маньяк. Противоположность автоответчика, игрок, придерживающийся данной стратегии, повышает ставку с большим спектром рук.
* ABC-регуляр. Сбалансированная стратегия, которая подразумевает как блеф, так и розыгрыш уверенных рук.

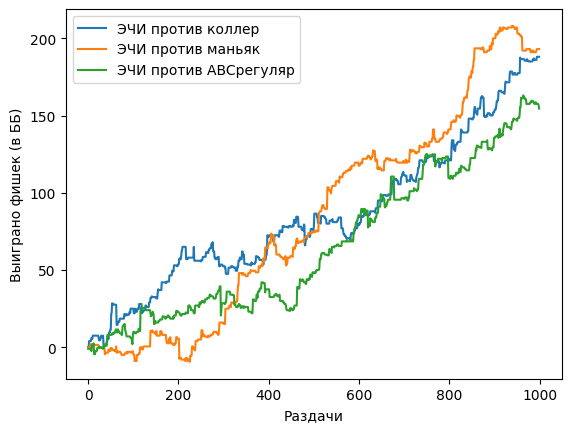
Результаты игр моделей АК и ЭЧИ против базовых стратегий представлены в таблице ниже:

**Табл 1.** Средний выигрыш за раздачу обученных моделей против базовых стратегий.

|  |  |  |  |
| --- | --- | --- | --- |
| Снизу – тестируемая, справа – тестирующая политика | Автоответчик | Маньяк | ABC-регуляр |
| АК | 2.82 | 0.84 | 0.65 |
| ЭЧИ | 0.19 | 0.19 | 0.15 |



**Рис. 9.** график суммарного выигрыша модели актёр-критик.



**Рис. 10.** график суммарного выигрыша модели, обученной на игре с применением эмуляции человеческой игры(ЭЧИ).

Из полученных результатов можно сделать вывод, что обе обученные модели показали свою эффективность против базовых стратегий. Рассмотрим каждую из отдельно

У модели актёр-критик все графики выигрышей отличаются, конечно это из-за специфики стратегии противников. Разные противники по-разному реагируют на действия модели, а также модель по-разному реагирует на действия противников.

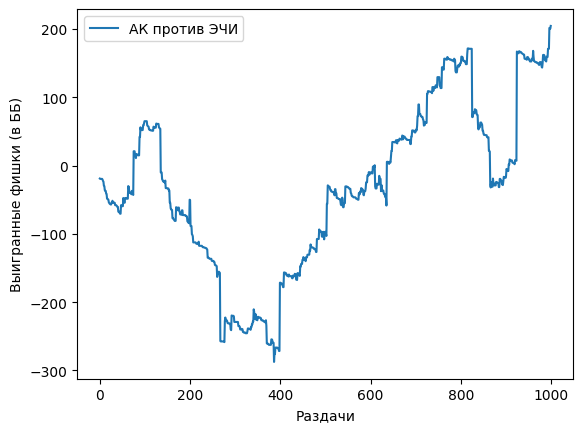
У модели, обученной на эмуляции человеческой игры (ЭЧИ), ситуация немного другая. Из графиков можно сделать вывод, что данная модель в любой ситуации модель играет очень сдержанно и осторожно. С одной стороны, это хорошо, так как модель в плюсе, но с другой, актер-критик во всех случаях выиграл больше.

* 1. *Результаты игр моделей между собой.*

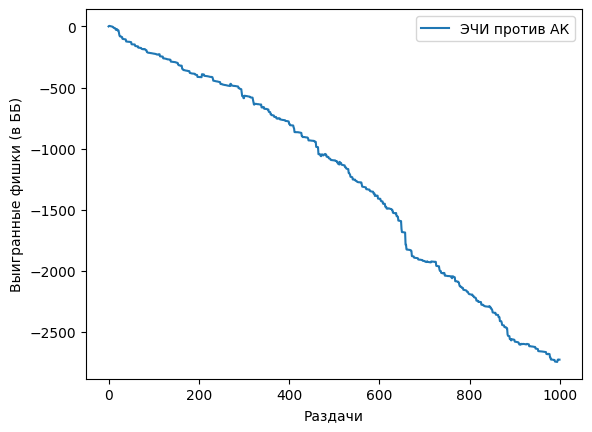
Теперь чтобы определить какая из двух обучаемых моделей лучше рассмотрим два случая: АК модель против ЭЧИ моделей и ЭЧИ модель против АК моделей.

**Табл 2.** Средний выигрыш за раздачу моделей при игре против друг друга.

|  |  |  |
| --- | --- | --- |
| Снизу – тестируемая, справа – тестирующая политика | АК | ЭЧИ |
| АК |  | 0.2 |
| ЭЧИ | -2.7 |  |



**Рис. 11.** график суммарного выигрыша модели актёр-критик против моделей ЭЧИ.



**Рис. 12.** график суммарного выигрыша модели актёр-критик против моделей ЭЧИ.

После просмотра графиков напрашивается очевидный вывод, лучшей из рассмотренных моделей является модель актёр-критик.

При игре против 5 моделей ЭЧИ основные потери фишек у актёр-критика – это неудачные олл-ины, но в целом у АК наблюдается стабильное увеличение общего выигрыша.

1. **Заключение**

Результатам данной статьи оказалось, что самым оптимальным методом изучения и нахождения эффективной политики действий, как в многопользовательской игре в покер, так и в других подобных средах где информация неполная, является обучение с подкреплением, а точнее модель актер-критик. Обучение с подкреплением приводит к тому, что процедура разработки политики самостоятельно воспроизводится с нуля без каких-либо данных о правилах, по которым среда существует, или экспертных навыков. И данная работа показала, что такой подход эффективнее, чем противоположный подход.

И одно из самых главных преимуществ модели, представленной в данной работе. Модель, обученная вышеописанным способом, может реагировать в режиме реального времени.

И последнее, но столь же важное преимущество: методы обучения с подкреплением, основанные на модели актер-критик, могут быть применены в других средах с несовершенной информацией. Например, в финансовой индустрии, где инвесторы постоянно принимают решения на основе неполной информации о рынке и поведении других игроков, такие алгоритмы могут быть использованы для прогнозирования трендов и определения наилучших инвестиционных стратегий. Кроме того, подобные алгоритмы могут использоваться для разработки новых решений в области кибербезопасности, где анализ поведения злоумышленников и моделирование их действий могут быть полезными в принятии соответствующих мер по защите целей.

# Список литературы:

1. M. Johanson, N. Burch, R. Valenzano, M. Bowling, Evaluating State-Space Abstractions in Extensive-Form Games // International Conference on Autonomous Agents and Multiagent Systems. 2013. c. 271–278.
2. S. Ganzfried, T. Sandholm, Potential-Aware Imperfect-Recall Abstraction with Earth Mover’s Distance in Imperfect-Information Games // AAAI Conference on Artificial Intelligence. 2014. С. 682–690.
3. N. Brown, S. Ganzfried, T. Sandholm, A Champion Two-Player No-Limit Texas Hold’em Poker-Playing Program // International Conference on Autonomous Agents and Multiagent Systems. 2015. с. 7–15
4. S. Srinivasan, M. Lanctot, V. Zambaldi, J. Pérolat, K. Tuyls, R. Munos, M. Bowling, Actor-Critic Policy Optimization in Partially Observable Multiagent Environments // International Conference on Autonomous Agents and Multiagent Systems. 2012. C. 837–846.
5. Лаврик А.Е, Тропченко А.А, Cравнение методов агентного моделирования на примере игры в разновидность покера «техасский холдем» // Международный научно-исследовательский журнал. 2021. С. 75-78
6. M. B. Johanson, Robust strategies and counter-strategies: from superhuman to optimal play, Ph.D. thesis, University of Alberta (2016).
7. D. Billings, A. Davidson, J. Schaeffer, D. Szafron, The challenge of poker // Artif. Intell. 2002. C. 201–240
8. N. Brown, T. Sandholm, Superhuman AI for heads-up no-limit poker: Libratus beats top professionals // Science 2018. C. 418–424
9. Julien Perolat, Bilal Piot, Olivier Pietquin. Actor-critic fictitious play in simultaneous move multistage games // International Conference on Artificial Intelligence and Statistics. 2018. C. 919–928
10. A. Rubinstein, Inapproximability of Nash Equilibrium // SIAM J. Comput. 2018. C. 917–959
11. M. A. Zinkevich, M. Bowling, M. Wunder, The lemonade stand game competition: Solving unsolvable puzzles // ACM SIGecom Exchanges. 2011. C. 35–38
12. M. Moravčík, M. Schmid, N. Burch, V. Lisý, D. Morrill, N. Bard, T. Davis, K. Waugh, M. Johanson, M. Bowling, DeepStack: Expert-level artificial intelligence in heads-up no-limit poker // Science. 2017. C. 508–513
13. Brown, N, Sandholm, T. Superhuman AI for multiplayer poker // Science. 2019. C. 885–890
14. N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, Model-based deep learning // arXiv:2012.08405. 2020
15. Wong, A.; Bäck, T.; Kononova, A.V.; Plaat, A. Deep Multi-Agent Reinforcement Learning: Challenges and Directions // Artif. Intell.Rev. 2023. C. 5023–5056.
16. 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>