**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ   
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа бакалавриата «Прикладная математика и информатика»

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Руководитель проекта,  ООО «Яндекс. Технологии», разработчик  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. А. Симагин  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. | УТВЕРЖДАЮ  Академический руководитель образовательной программы  «Прикладная математика и информатика»  доцент, канд. физ.-мат. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. С. Конушин  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2019 г. |
| |  |  | | --- | --- | | Подп. и дата |  | | Инв. № дубл. |  | | Взам. Инв. № |  | | Подп. и дата |  | | Инв. № подл. | RU.17701729.502830-01 81 01-1 | | **ГЛУБОКОЕ ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ В ИГРЕ РЕНДЗЮ.**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.502830-01 81 01-1-ЛУ**  Исполнитель  студент группы 174 ПМИ  \_\_\_\_\_\_\_\_\_\_\_/Кузнецов Д.С./  «\_\_\_»\_\_\_\_\_\_\_\_\_\_2019 г.  **2019** | |  |

УТВЕРЖДЕН

RU.17701729.502830-01 81 01-1-ЛУ

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Подп. и дата |  | | Инв. № дубл. |  | | Взам. Инв. № |  | | Подп. и дата |  | | Инв. № подл. | RU.17701729.502830-01 81 01-1 | | **ГЛУБОКОЕ ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ В ИГРЕ РЕНДЗЮ.**  **Пояснительная записка**  **RU.17701729.502830-01 81 01-1**  **Листов 12**  **2019** |  |

# 

Оглавление

[1. ВВЕДЕНИЕ 4](#_Toc1110509)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 5](#_Toc1110510)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 6](#_Toc1110511)

[3.1. Постановка задачи на разработку программы 6](#_Toc1110512)

[3.2. Описание алгоритмов и функционирования программы 6](#_Toc1110513)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 8](#_Toc1110514)

[4.1. Ориентировочная экономическая потребность 8](#_Toc1110515)

[4.2. Предполагаемая потребность 8](#_Toc1110516)

[4.3. Преимущества разработки по сравнению с существующими аналогами 8](#_Toc1110517)

[5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ 9](#_Toc1110518)

[Приложение 1 10](#_Toc1110519)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 12](#_Toc1110520)

# ВВЕДЕНИЕ

* 1. **Наименование**

«Глубокое обучение с подкреплением в игре рендзю».

* 1. **Документы, на основании которых ведется разработка**

Приказ от 12.12.2018 г. № 2.3-02.1212-02 «Об утверждении тем, руководителей и консультантов курсовых работ студентов образовательной программы Прикладная математика и информатика факультета компьютерных наук» декана факультета компьютерных наук НИУ ВШЭ.

# НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

**2.1 Функциональное назначение**

Программа предназначена для предоставления игровой среды «рендзю» пользователю и для предоставления машинного интеллекта [Приложение 1], способного противостоять пользователю в игре.

**2.2 Эксплуатационное назначение**

Программа предназначена для разыгрывания партий игры «рендзю» с машинным интеллектом.

# ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

## Постановка задачи на разработку программы

Разрабатываемая программа должна предоставлять пользователю следующие возможности:

1. Предоставлять пользователю интерфейс по средствам консоли для взаимодействия с игровым движком.
2. Предоставлять пользователю запуск новой партии без перезагрузки интерфейса
3. Предоставлять пользователю репрезентативный интерфейс, т.е. интерфейс программы должен корректно отображать состояние текущей партии игры.
4. Предоставлять пользователю возможность соревноваться с машинным интеллектом, обученным играть в «рендзю».

Программа должна:

1. Иметь собственную имплементацию игрового движка.
2. Иметь собственную имплементацию машинного интеллекта искусственного оппонента.
3. В режиме «разработки» обучать машинный интеллект оппонента.

## Описание алгоритмов и функционирования программы

* + 1. **Консольный интерфейс**

Консольный интерфейс должен предоставлять пути взаимодействия с программным комплексом для запуска очередной партии игры и изменения состояния партии в соответствии с правилами игры «Рендзю».

Интерфейс представляет из себя единственное активное состояние – отрисовка текущего состояния партии и запрос очередного хода.

Запуск программного комплекса с помощью консольного интерфейс осуществляется запуском главного исполняемого файла (sh-script renjuRL) в корне директории программы. После запуска исполняемого файла открывается интерфейс взаимодействия с игровым процессом с помощью строчных литералов [Примечение 1].

* + 1. **Структура и обучение машинного интеллекта**

Машинный интеллект – это не что иное, как нейронная сеть (далее Policy NN). Задача Policy NN состоит в построении распределения на доступных ходах, пропорционального убежденности сети в том, какой ход ведет к победе, для заданного состояния игры (доски).

Вход нейронной сети – это матрица игры, представленная в определенном виде [Приложение 1]. Выход нейронной сети – это вектор распределения, где i-ая компонента – убежденность (отнормированная) сети в том, что i-ый ход (ход в (i / 15, i mod 15) клетку) ведет к победе.

Архитектура нейронной сети представлена в [Приложение 1].

Обучение Policy NN производилось на многотысячной выборке партий профессиональных игроков в «Рендзю». Стоит заметить, что партии из обучающей выборки производились при соблюдении принятых в профессиональной среде регламентов. Задача данной нейронной сети обучится играть без заданных регламентов. Оставление исходной выборки без изменения гарантирует, что после успешного обучения нейронной сети будет свойственно избегать запрещенные регламентом ходы. Для решения этой проблемы используется аугментация выборки, в частности случайное линейное движение выпуклой оболочки (минимальной площади) ненулевых (занятых) клеток доски.

Метрика качества, используемая при обучении, - categorical accuracy, функционал ошибки – log loss. Оптимизация функционала производилась по средствам Adam алгоритма.

* + 1. **Алгоритм функционирования программного комплекса**

После запуска консольного интерфейса и игрового движка (процесс описан в п. 3.2.1) пользователю представляется возможность выбрать цвет фишек (очередность его ходов). После выбора начинается игра. Друг за другом ходы выполняют пользователь и машинный интеллект в соответствии с их цветом фишек. Очередная пара ходов выполняется по следующему плану.

После команды игрока (move x y [Приложение 1]) игровой движок проверяет корректность хода: занята ли клетка, не выходят ли координаты за границы доски, действительно ли сейчас ход пользователя. В случае корректности хода фишка устанавливается на доску и запускается процесс сканирования выигрышных комбинаций: ищутся по всей доске всевозможные расстановки пяти фишек в ряд. Если такая расстановка найдена, игра завершается и выводится сообщение о победе текущего игрока, в противном случае ход передается противнику.

Когда очередь доходит до машинного интеллекта, состояние доски передается в виде матрицы Policy NN. Внутри класса Policy NN исходная доска преобразуется к требуемую формату [Приложение 1], затем программный комплекс запрашивает нейронную сеть за выдачу предсказания. Следующий ход оппонента выбирается, как ход в первую корректную клетку в порядке сортировки по убыванию убежденности НС в ходах.

Игра итеративно продолжается, пока не будет достигнута выигрышная комбинация одним из игроков или пока не закончатся корректные ходы у обоих игроков. В последнем случае объявляется ничья.

# ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

## Ориентировочная экономическая потребность

Программа удовлетворяет две экономические потребности целевой аудитории:

1. Развлечение
2. Тренировка профессиональных навыков

Благодаря наличию машинного интеллекта в программе, пользователь имеет возможность играть без участия сторонних пользователей. Игру «рендзю» можно рассматривать с двух перспектив: вид времяпрепровождения и профессиональная деятельность. Машинный интеллект обучен играть на профессиональном уровне, поэтому удовлетворяет потребность пользователя в развитии навыков игры по средствам состязания с описываемым машинным интеллектом. Развлекательная потребность удовлетворяется данным программным комплексом аналогично с тем лишь исключением, что в процессе игры не преследуется тренировка навыков игры.

## Предполагаемая потребность

Подавляющее большинство современных реализаций данной задачи имеют машинного оппонента с низким уровнем игры, не достаточного для тренировок на профессиональном уровне. Благодаря особому процессу обучения машинного интеллекта, данный программный продукт способен удовлетворить эту пользовательскую потребность.

## Преимущества разработки по сравнению с существующими аналогами

1. Высокий уровень игры машинного интеллекта.
2. Наличие консольного интерфейса

# ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. Статья «Mastering the game of Go with deep neural networks and tree search» [Электронный ресурс]//

URL: https://github.com/dasimagin/renju/blob/master/AlphaGo.pdf (Дата обращения: 15.02.2019, режим доступа: свободный)

Приложение 1

**Применяемые алгоритмы и термины**

**Машинный интеллект**

**-** компьютерная имплементация алгоритма статистического решения некоторой задачи на новых данных на основе выявления закономерностей в известных данных в прошлом.

**Игровой движок**

- базовое программное обеспечение компьютерной игры, осуществляющее обработку событий взаимодействия с игровой средой, обработку событий изменения среды, поддержку информацию о состоянии среды.

**Взаимодействие с игровым движком с помощью строковых литералов**

Процесс взаимодействия с игровым движком начинается с приглашающего литерала “$: ” игровым движком для пользователя на выходной поток. Данный литерал означает, что игровой движок готов принимать команды от пользователя.

Доступные команды взаимодействия:

* move x y

Устанавливает фишку на позицию (x, y), где x – ордината координатной плоскости, y – обсцисса. Отчет начинается с левого верхнего угла игровой площадки. Положительные лучи направлены вниз и вправо, соответственно, для ординаты и обциссы.

* show

Вывод состояние игрового поля

* finish

Завершение процессов игры

**Формат входной выборки Policy Network**

Элемент входной выборки представляет из себя трехмерный тензор. Рассмотрим отдельно каждую ось тензора.

Первая ось – матрица ходов первого игрока. Матрица заполнена нулями кроме клеток, куда сходил игрок, играющий за черные фишки, последние заполнены единицами.

Вторая ось – матрица ходов второго игрока. Матрица заполнена нулями кроме клеток, куда сходил игрок, играющий за белые фишки, последние заполнены минус единицами.

Третья ось – матрица занятых клеток. Матрица заполнена индексом игрока, чей ход следующий, (1 или -1) во всех клетках кроме клеток, куда сходил либо первый либо второй игрок, последние заполнены нулями.

Данный формат входных данных выбран не случайно. Он позволяет избежать отдельного обучения двух сетей: для черных и для белых. В случае совместной матрицы ходов (ставим фишки черных и белых в одну матрицу) нейронная сеть не способна на этапе обучения детерминировать чей ход следующий и за кого необходимо предсказывать оптимальный ход.

**Архитектура Policy Network**

Нейронная сеть представляет из себя два последовательных блока: блок сверточно-субдискретизирующий и полносвязный блок.

Сверточно-субдискретизирующий (Conv-Pool layers) представляют из себя совокупность сверточных и субдискретизирующих слоев. Данная Policy NN имеет 3 последовательных сверточных слоя из 128 фильтров размерностей 3х3, 4х4, 5х5, начиная с первого слоя. После каждого слоя добавлена нелинейность в лице ReLu функции и добавлен слой батч-нормализации. Завершает данный блок субдискретизирующий слой с функцией максимума.

Полносвязный блок представляет из себя два последовательных скрытых полносвязных слоя из 500 нейронов на каждом слое с сигмоидальной функцией активации. После каждого слоя добавлен dropout слой для избежания переобучения. Выходом нейронной сети является – полносвязный слой с нелинейностью softmax.

Два блока соединены последовательно. Выход сверточно-субдискретизирующего слоя – вход полносвязного слоя.

В данной задаче важно использовать в первом блоке фильтры относительно высокого разрешения. Фильтры 2х2 не дают практически никакого качества, т.к. по квадрату 2х2 сложно произвести близкое к биективному отображение в множество комбинаций, т.е. по квадрату 2х2 сложно выделить выигрышные паттерны.

# ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| измененных | замененных | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |