

**Применение технологий машинного обучения в разработке
симулятора шахмат «Гала»**

Сорокин Д.А.¹, Огнева М.В.²

¹nidernider001@gmail.com, ²ognevamv@mail.ru

Саратовский государственный университет имени Н.Г. Чернышевского

Аннотация. В статье рассказывается о самостоятельно разработанном приложении, реализующем исторический вариант шахмат «Гала», а также о процессе разработки компьютерного игрока с применением технологий машинного обучения в рамках этого приложения.

Ключевые слова: шахматы, игра-симулятор, машинное обучение, самообучение.

Компьютерные, мобильные и другие игры в электронном формате уже долгое время являются неотъемлемой частью современной культуры. Приложения на самых разнообразных платформах глубоко вошли в жизнь каждого человека.

Новейшие технологии позволили перенести многие традиционные настольные логические игры в электронный формат. Одной из наиболее известных игр подобного жанра являются шахматы, имеющие огромное значение в развитии детей и подростков. В сети интернет можно найти большое количество шахматных движков – программ, способных играть в шахматы при помощи анализа доступных ходов. К шахматным движкам можно отнести такие программы как Stockfish, AlphaZero, Leela Chess Zero и другие. Каждая программа имеет свои уникальные особенности, в число которых входит способ реализации компьютерного игрока.

Сложности в разработке исторического варианта шахмат

В противоположность обычным шахматам, выбранная для разработки игра «Гала» является в наше время малопопулярной и редкой, что вызвало соответствующие трудности при поиске материала по ней. Согласно сведениям, полученным на портале энтузиастов традиционных настольных игр www.cynningstan.com и отдельно опубликованном документе за их авторством, информация об игре «Гала» (или «фермерских шахмат») сохранилась до настоящего времени благодаря Роберту Чарльзу Беллу – исследователю настольных игр, описавшему игру в своей книге «The Boardgame Book»[1]. Получить доступ к копии книги не представилось возможным, вместо этого был осуществлен анализ различных интернет-ресурсов, посвященных данному виду шахмат. Используемые в собственном приложении правила игры были основаны на результатах этого анализа.

Обзор разработанного приложения

Игра-симулятор шахмат «Гала» создана на кафедре информатики и программирования. Приложение разработано на языке C# с использованием интерфейса программирования приложений Windows Forms. Игра является пошаговой и рассчитана на двух игроков. В роли оппонента выступает компьютерный противник, также существует режим игры компьютера против компьютера. В отличие от обычных шахмат, основным условием победы является перемещение двух союзных королей на четыре центральные клетки игрового поля. Взятие всех королей соперника также приводит к победе, а потеря одной из фигур обеими игроками означает ничью.



Рис. 1. Окно приложения

Шахматная доска имеет размер 10 на 10 клеток и условно разделяется на пять зон. Первые четыре угловые зоны смежны с каждым из углов шахматной доски. Последняя центральная зона представляет собой крест с шириной линий в две клетки. Поведение фигур может изменяться, если они попадают в центральную зону. Пешки в союзных угловых зонах могут перемещаться и брать вражеские фигуры только по диагонали в сторону центра доски, но при выходе из них получают возможность ходить и брать фигуры в любом направлении. Перемещение в центральную зону инвертирует поведение ладей и слонов (ладья начинает ходить как слон и наоборот).

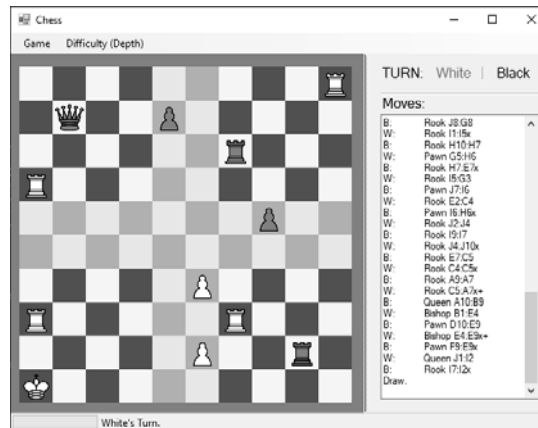


Рис. 2. Завершение игры через ничью

Некоторые фигуры также имеют особые варианты хода. Две центральные пешки в каждой из угловых зон игрового поля могут переместиться сразу на две клетки вперед на своем первом ходу. Короли имеют возможность переместиться с четырех центральных клеток шахматной доски на любую незанятую позицию, которая была свободна в начале игры.

Компьютерный игрок на основе технологии машинного обучения

Компьютерные игроки с применением технологий машинного обучения имеют долгую историю. Следует учитывать, что большинство из них использует различные формы готовых знаний об игре: примеры, своды правил, оценщики позиций, системы весов и т.д. Эти настройки позволяют алгоритмам быть эффективными игроками в рамках различных игр. Но они также способны и лимитировать дальнейшее развитие алгоритмов, распространяя на них сформированные человеком предвзятости.

Данная проблема была решена в рамках шахматного движка AlphaZero, который не использует для обучения какие-либо готовые данные. Более того, он изначально не знает об игре, в которую будет играть, ничего кроме набора доступных во время очередного хода действий. На текущий момент AlphaZero является одним из сильнейших алгоритмов, способным играть в традиционные шахматы[2]. Согласно описанию алгоритма, представленного разработчиками, вместо созданной вручную оценочной функции AlphaZero использует глубокую нейронную сеть. Нейронная сеть принимает на вход состояние текущего игрового поля, затем возвращает вектор распределения вероятностей, длина которого равна количеству возможных действий на игровом поле, и

скалярное значение, которое является оценкой ожидаемого окончания игровой партии: -1 в случае поражения конкретного игрока, 1 – в случае победы, небольшое положительное значение – в случае ничьи. Данные вероятности для ходов рассчитываются в процессе самообучения.

Вместо механизма альфа-бета отсечения AlphaZero использует алгоритм Монте-Карло поиска по дереву (MCTS) общего назначения. Каждый поиск состоит из серии смоделированных игр, которые проходят по дереву от корня (начального состояния игрового поля) до листьев (окончаний игры). Каждая симуляция проходит путем выбора для очередного состояния игрового поля оптимального варианта хода. На выходе алгоритма получается вектор распределения вероятностей возможных вариантов хода для начального состояния игрового поля. Подробнее с принципом работы алгоритма можно ознакомиться в источнике[3].

Для реализации аналогичного алгоритма в собственном приложении используется PyTorch – фреймворк с открытым исходным кодом, предназначенный для работы с технологиями машинного обучения. Функционал фреймворка описан на его официальном сайте pytorch.org. Фреймворк PyTorch работает с функциями и библиотеками языка Python, однако игра-симулятор «Гала» была разработана на языке C#. Для решения проблемы совместимости была использована разработанная пользователями библиотека TorchSharp, которая позволяет использовать функции фреймворка PyTorch в среде разработки VisualStudio. Код данной библиотеки и её спецификация находятся в открытом доступе на веб-сервисе GitHub[4]. Функционал фреймворка позволяет задавать нейронную сеть с произвольным количеством слоев разного типа.

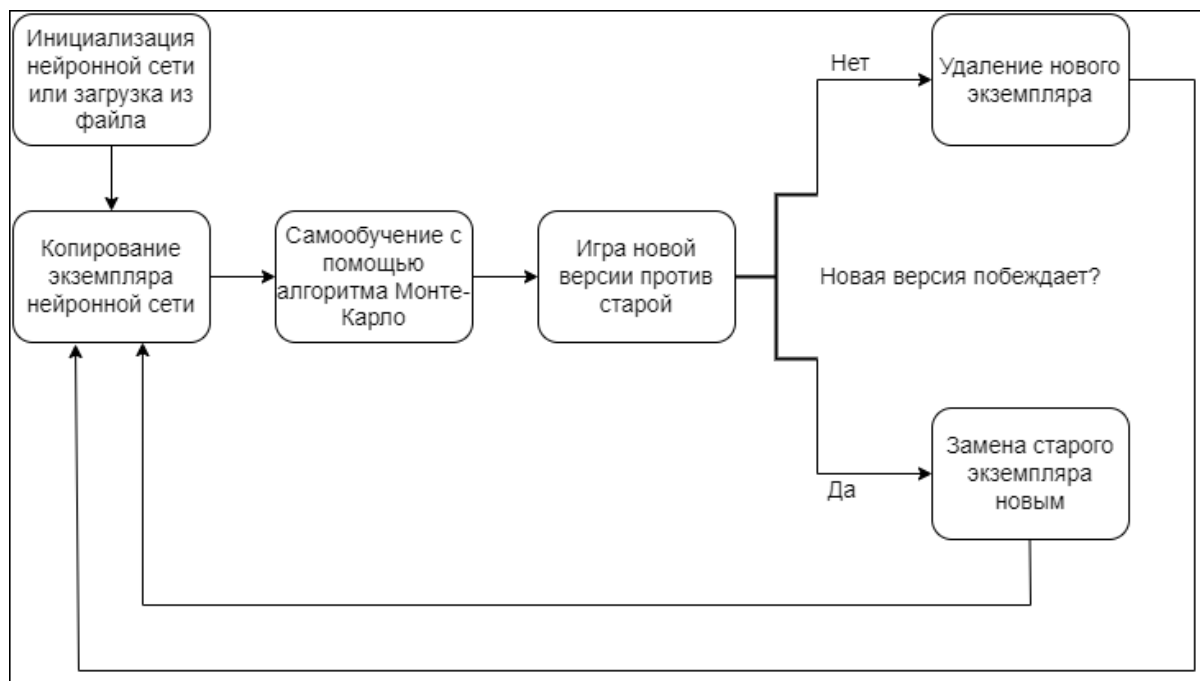


Рис. 3. Схема работы алгоритма самообучения

Уникальный набор правил и необычный формат игры позволят привлечь игроков подросткового возраста, повышая их заинтересованность как в шахматах, так и в настольных играх в целом. В рамках дальнейшей работы над приложением планируется осуществить настройку и тренировку алгоритма. Затем планируется оценка его уровня игры в сравнении с традиционным алгоритмом на основе альфа-бета-отсечения, уже реализованного в программе (подобный алгоритм использует шахматный движок Stockfish), и размещение готового приложения в общий доступ.

Список литературы

- [1] Bibliography of Traditional Board Games [Электронный ресурс] URL: https://www.academia.edu/15246914/Bibliography_of_Traditional_Board_Games (дата обращения 17.09.2022).
- [2] *Sadler M. Game changer* / M. Sadler, N. Regan. – Alkmaar, The Netherlands: New in Chess, 2019. – 414 p.
- [3] Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm [Электронный ресурс] URL: <https://arxiv.org/abs/1712.01815> (дата обращения 22.09.2022).
- [4] Dotnet/TorchSharp [Электронный ресурс] URL: <https://github.com/dotnet/TorchSharp> (дата обращения 26.09.2022).